April 1983
USA $2.95 (UK£2.00)

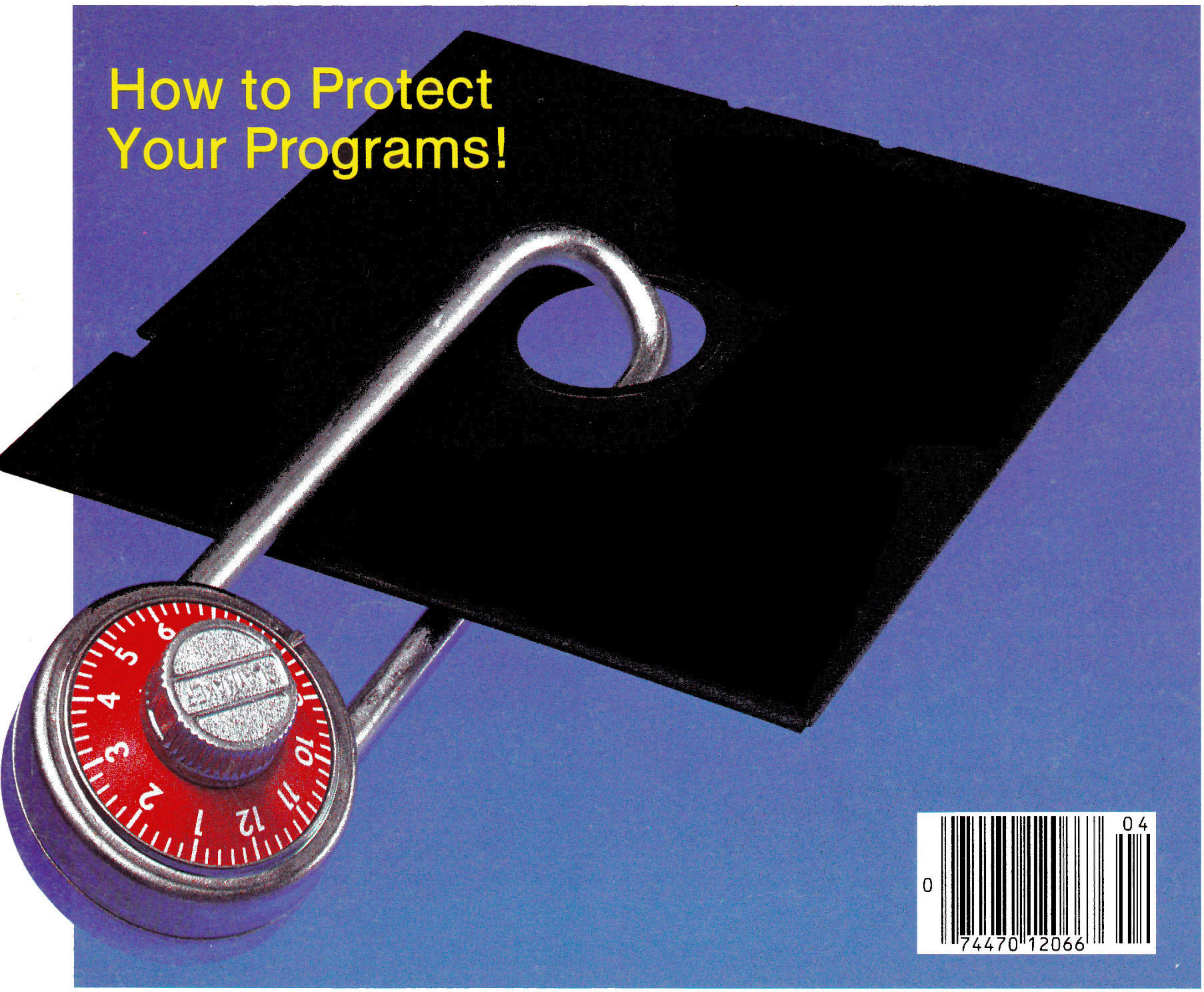
Logo Gets Down to Business

inCider™

Green's Apple* Magazine

IIe Software... What Works?

Join the Action
Strategy and Arcade Games





SPECIALLY TARGETED. Works in harmony with your z-card/cpm equipped Apple II. No more floppy shuffle. Fast one pass compiler that doesn't generate unnecessary intermediate files which require further assembly.

PROFESSIONALLY DESIGNED. Crafted by a project team with over 15 years of Pascal programming experience. A powerful Pascal that includes all the features necessary for professional program development including random files and dynamic strings.

INCREDIBLE SPEED. Resultant stand alone programs execute 10-20 times faster than the equivalent Basic or up to 4-1/2 times faster than U.C.S.D. programs.

	TRADEMARK
CPM	Digital Research
Apple	Apple Computers
UCSD	University Calif. San Diego
Alcor Pascal	Alcor Systems

POWERFUL TEXT EDITOR. Has Macro capability, reconfigurable and understands smart terminals.

EXCEPTIONAL DOCUMENTATION. Includes 250 page documentation pkg. that guides you step by step.

OPTIONAL ADVANCED DEVELOPMENT PKG. Includes Optimizer and Native Code Generator for even faster executing programs. ADP: $125 + shipping.

ALCOR PASCAL: A new standard of excellence from the professionals at Alcor Systems.

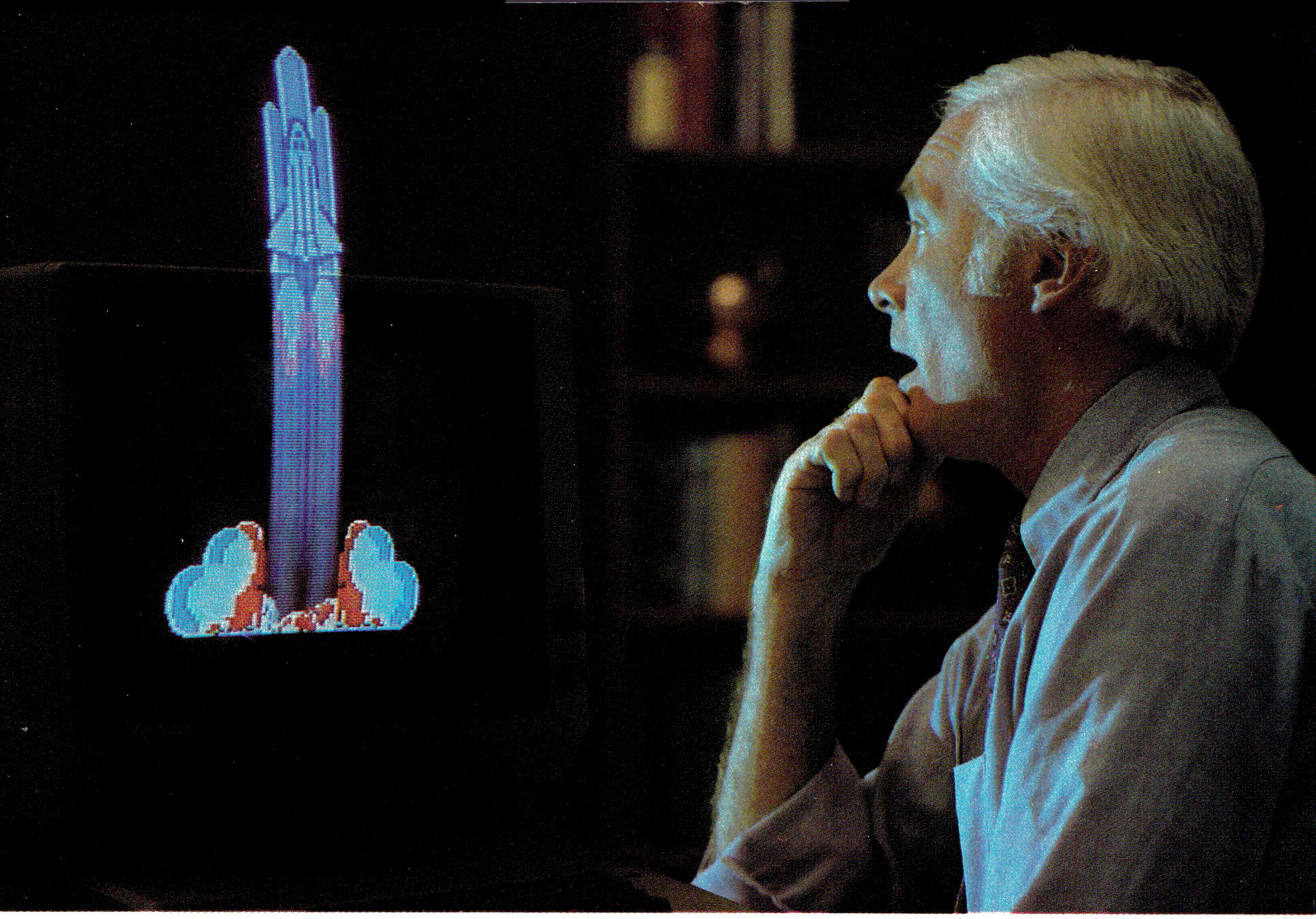
CALL OR WRITE for a free 20 page technical brochure with all the facts. 214-226-4476. Pascal System: $199 + shipping.

ALCOR SYSTEMS

Alcor Systems
800 W. Garland Ave.
Garland, Texas 75040

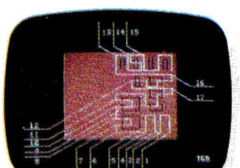
Dealer Inquiries Invited

Circle 1 on Reader Service card.



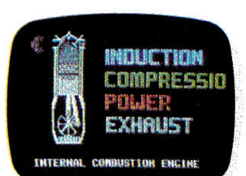
THE GRAPHIC SOLUTION

Solve your toughest communication problems with the Graphic Solution™, a sophisticated, new graphics package from Accent Software.

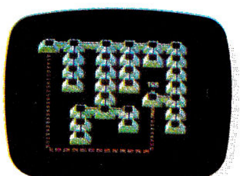


With precise, multi-speed ANIMATION create captivating sales presentations and product demonstrations that will both intrigue and inform your clients and customers. Watch their reactions; you'll see your messages getting through.

Develop educational materials and training aids that MIX TEXT AND GRAPHICS on the screen, breathing new life into abstract, hard-to-grasp concepts. Mix programs too. Images can be displayed on backgrounds loaded from any of your other programs. Construct custom TYPEFACES AND TYPESIZES to balance the visual elements.



Tired of run-of-the-mill business graphics? Change standard charts and graphs into colorful THREE DIMENSIONAL PERSPECTIVES. Add text and animate the data to show the relative rates of change for your most important information. Like cash flow projections. Or revenue estimates.



Plot flowcharts, time and motion studies, industrial process flows with COLOR-CODED ELEMENTS highlighting critical paths. Animate the sequences to show how flows actually progress.

Work with live action? Prepare film and videotape storyboards using the unique FRAME-BY-FRAME graphic sequencer that lets you create and animate a video story before shooting.

Whatever your graphic communication demands—in the business world, the arts, industry, education—The Graphic Solution™ at $149.95 has the answer. Take a hard look at The Graphic Solution. You'll like what you see.



The Graphic Solution requires a 48K Apple II with ROM Applesoft and DOS 3.3.

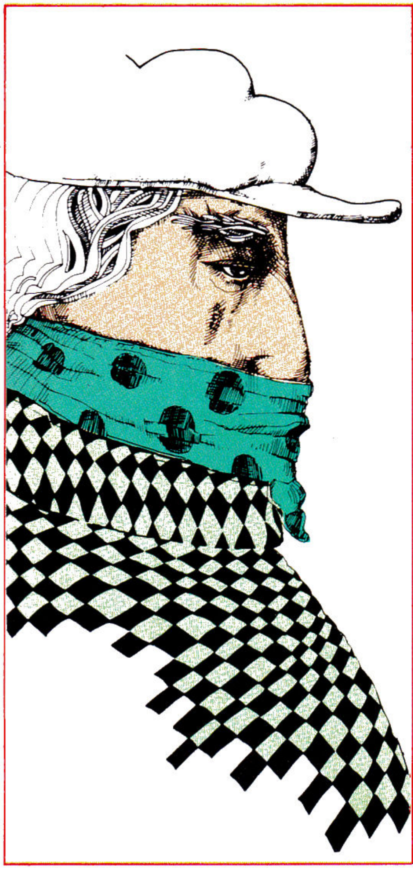
See your local dealer or send $10.00 for a demonstration diskette to:

ACCENT SOFTWARE, INC.

3750 Wright Place, Palo Alto, Calif. 94306 Telephone 415-856-6505

Circle 251 on Reader Service card.

Apple is a registered trademark of Apple Computer, Inc.



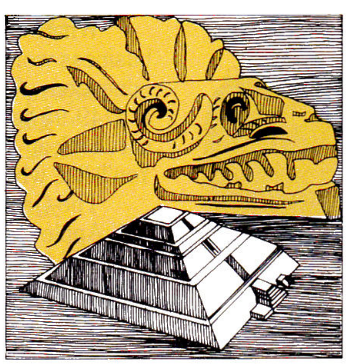
The Apple Combination Lock, p. 106

ARTICLES

To everyone who enjoyed Joe Magee's "EPROM Edification" in last month's issue: Watch for a follow-up in May that will show you how to add the ability to read EPROMs into Apple memory.

DEPARTMENTS

Cover photo by Hyam Siegel.



Aztec review, p. 152

inCider (ISSN pending) is published 12 times a year by 1001001 Inc., 80 Pine St., Peterborough, NH 03458. Phone: 603-924-9471. Second class postage pending at Peterborough, NH, and additional mailing offices. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada and Mexico, $27.97—one year only, U.S. funds drawn on a U.S. bank. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. BC Canadian distributor: Graymar Data Services, Ltd., #4 258 E. 1st Ave., Vancouver, BC V5T 1A6. Foreign subscriptions (surface mail), $44.97—one year only, U.S. funds drawn on a U.S. bank. Foreign subscriptions (air mail), please inquire. In South Africa contact *inCider*, P.O. Box 782815, Sandton, South Africa 2146. All U.S. and Canadian subscription correspondence should be addressed to *inCider*, Subscription Department, P.O. Box 911, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send address changes to *inCider*, Subscription Services, P.O. Box 911, Farmingdale, NY 11737.

Remarks From the Publisher... Wayne Green



The e Is Here, And Lisa Too!

Many of us were starting to look for Volkswagen-type ads for the Apple II that go something like, "How Long Can We Keep Handing You This Old Line?" After all, the II ripened in 1977, *six* years ago, which is about two generations in computer lifetimes. Considering the rapid developments in technology, that is an amazing life for a high tech product.

Yet, when I think about it, I have to admit that computers are radically different from other products. Their total dependence upon a combination of software and user skill gives them a far longer life than is possible for a hardware product alone. Indeed, just look at the changes we've seen in other high tech products during this same period. Watches, under pressure from Casio, have changed considerably, as have calculators, radios, stereos, and so on. The Apple II has been changing too, but it has been a low visibility change—one brought about by growth in software and by information on how to use the system.

We face a serious drawback when hardware makes a big change. Many or all of the hard-won gains in programs and user information—which thousands of patient (and impatient like me) people have supported—are lost.

Radio Shack has been ruthless in this regard. One gets the strong impression that Radio Shack begrudges any product that they don't make money on. I've talked with Tandy insiders who have put it just that bluntly.

One of the great benefits that man has over most other living things is time binding—the ability to pass along learned lessons to each new generation. We have this in consistent generations of computers, and it is of inestimable value.

With Radio Shack, we have to go back almost to square one with each new system, losing untold man hours of programs and information on how to use the system. With the Apple we've had this long buildup of information and I think that is more at the heart of the success of the Apple than any other single factor.

It may have been fortunate in the long run that the Apple III bombed. It was modestly compatible with the II, yet for those of us who have been using the III and the II, the improvements achieved seem minor compared to the value we find in the II, as enhanced by hundreds of magazine articles on its use and thousands of programs enabling it to do a wide range of chores.

"It is unfortunate that many computer stores do not yet appreciate the values of the Apple."

Perhaps I wasn't as enthusiastic as many of the Apple people about the long awaited unveiling of their new models. And I'm not dying to see the promised Macintosh model, unless it is 100 percent compatible with Apple II software and won't throw away all that we've taken months (and even years) to learn about using the II.

It is unfortunate that many computer stores do not yet appreciate the values of the Apple and have been emotionally discounting it because of its age. This has led to price discounting, which has been devastating. Many Apple dealers have been actively discouraging Apple sales and pushing lines where they make substantially more money per sale. In this case the welter of Apple advertising has been working against the company.

Perhaps it is time for Apple to take a serious look at the whole sales picture. They should recognize that selling the Apple concept to dealers is as important as selling it to the end customer. When prospective Apple buyers are smothered with dealer enthusiasm for some higher ticket computer, it is time to make a major push to get things back into kilter. It's possible that the $10,000 Lisa will help with this.

But dealers are a key to computer sales and should be honored as such. This means that they should be given all possible help in selling the product. They should have the information they need: video tapes to make up for undertrained salespeople, easily understandable brochures, help at local shows, reprints of good equipment reviews, and a place to call for technical, sales, and delivery information. In *Selling Micros*, my dealer oriented magazine, I recently published a list of 13 important areas where a manufacturer should support dealers.

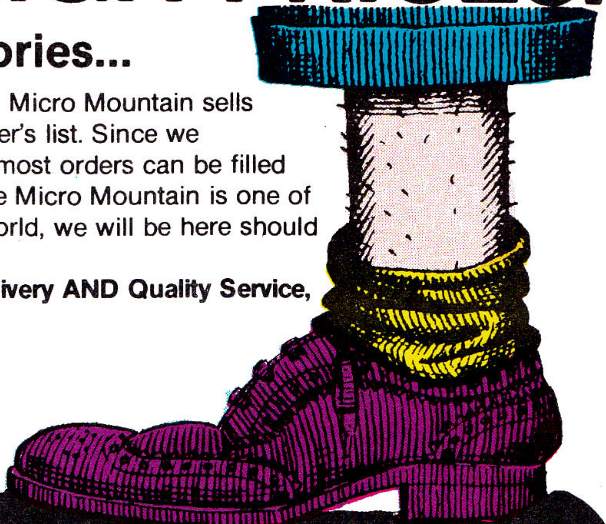
Making the Apple easier to use, via

STAMP OUT HIGH PRICES!

On Apple® Software & Accessories...

... without sacrificing on prompt delivery or quality service. Micro Mountain sells everything at prices substantially less than the manufacturer's list. Since we maintain such a large inventory (1000's of items in stock) most orders can be filled and shipped within 24 hours after you order. And, because Micro Mountain is one of the largest retail distributors of Apple® Software in the world, we will be here should you need service at anytime after you receive your order.

So, if you're interested in— **Low Prices, Prompt Delivery AND Quality Service, SHOP AT . . .**



micro mountain

NOW WE CARRY **FRANKLIN COMPUTERS** CALL FOR THE BEST PRICES!

OMEGA **Locksmith** $69.50	SILICON VALLEY **List Handler** $64.50
TAXAN — 12" AMBER **Monitor** $147.50	STRATEGIC SIMULATIONS **Bomb Alley** $47.50
EASTSIDE SOFTWARE **Wild Card** $109.50	HAYES **Micro Modem II** .. $277.50
MICRO LAB **Tax Manager** $147.50	DOW JONES **Market Analyzer** .. $275.50
KENSINGTON MICROWARE **Visicalc 3.3** $187.50	EDU-WARE **Alegebra IV** $26.50
VISTA **'Solo' Disk Drive** .. $279.50	VISTA **16K Ram Card** $59.50
SPINNAKER **Rhymes & Riddles** . $23.50	SILICON VALLEY SYSTEMS **Word Handler** $139.50
KENSINGTON MICROWARE **System Saver** $65.50	MICROSOFT **Multi Plan** $219.50

-ASK FOR FREE CATALOG-

EXTRA SAVINGS BONUS!

Any order of 3 or more programs entitles you to your choice of Original Adventure Game or Applesoft® Tutorial on disk FREE!

PHONE IN YOUR ORDER —TOLL FREE— (800) 854-5649

Washington State residents see phone numbers in Order Blank at Right.

ORDER PHONE HOURS:
Mon.-Fri. 9 to 6 (PST)
Sat. & Sun. 10 to 2 (PST)

BE SURE TO ADD $2.50 SHIPPING AND HANDLING FOR ALL SOFTWARE ORDERS, ADD ADDITIONAL $3.00 FOR BLUE LABEL (Air). SHIPPING ON HARDWARE ITEMS—EXTRA. Washington residents add 6½% sales tax. We accept MASTERCARD, VISA and AMERICAN EXPRESS. C.O.D.'s add $5.00.

NAME ____________________

STREET ____________________

CITY ____________ STATE & ZIP ____________

CARD # ____________ EXP. DATE ____________

SIGNATURE ____________________

14517 N.E. 169th ST., WOODINVILLE, WA 98072
ORDER PHONE — Outside Wash. — (800) 854-5649
Wash. Residents & Cust. Service (206) 483-2000

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

Circle 55 on Reader Service card.

PUBLISHER/EDITOR
Wayne Green
EXECUTIVE VICE PRESIDENT
Sherry Smythe-Green
GENERAL MANAGER
Debra Wetherbee
ASSISTANT TO PRESIDENT
Matt Smith
ASSISTANT PUBLISHER
Jeff DeTray
CORPORATE CONTROLLER
Roger Murphy
ASSISTANT TO CONTROLLER
Dominique Smith
CIRCULATION MANAGER
603-924-9471
Patricia Ferrante
BULK & NEWSSTAND SALES MANAGER
Ginnie Boudrieau

ADVERTISING, 603-924-7138
David Schissler, Manager;
Denis Duffy, Craig Everett: Sales;
Patty Lesser: Advertising Coordinator.
New England Advertising Representatives:
John A. Garland, Frank Surace,
Garland Associates Inc.,
Box 314 SHS,
Duxbury, MA 02332 617-934-6464

PRODUCTION
Nancy Salmon, Manager; Michael Murphy, Assistant; Frances Benton, Michael Ford, Phil Geraci, Donna Hartwell, Kimberly Nadeau, Lynn Parsons, Deborah Stone, Theresa Verville, Robert Villeneuve, Karen Wozmak; Ad Coordinators: Paula Ramsey, David Wozmak, Mary Seaver; Advertising Production: Bruce Hedin, Fiona Davies, Scott Philbrick, Jane Preston
PHOTOGRAPHY
Thomas Villeneuve, Supervisor; Katherine Coker, Sandra Dukette, Laurie Jennison, Irene Vail
TYPESETTING
Sara Bedell, Supervisor; Marie Barker, Melody Bedell, Michele DesRochers, Jennifer Fay, Prem Gongaju, Lynn Haines, Linda Locke, Nancy Wilson-Newman, Debbie Nutting, Lindy Palmisano, Susan Weller
DESIGN
Jonathan Graves, Creative Director; Christine Destrempes, Design Director; Joyce Pillarella, Supervisor; Denzel Dyer, Howard Happ, Susan Donohoe, Laurie MacMillan, Dion Owens, Diane Ritson, Susan Stevens, Donna Wohlfarth

the Lisa approach, is certainly a good idea. Few people have gotten the full benefit of programs such as VisiCalc because they are so complicated to learn and use. Computers should be made easier to use. Yet it's difficult for people—and that includes engineers—not to go with trends. The early Apple developments set a trend, with the switch from cassette to floppy disk. Yet, in perspective, Apple didn't do users any favors when they cut the cassette umbilical cord. The cost of a cassette recorder is small and of value to the computerist. Most of us have one around for other uses anyway.

Sure, cassettes are ineffective for storing data that is needed quickly. But then, they aren't very good as boat anchors either. Cassettes are almost ideal as a low cost medium for storing data and for programs which are not used much. And they are excellent as a medium for transferring programs. The cassette itself is much less costly than a disk. Its use cuts about $5 to $10 from the cost of a program—and that mounts up after a while! Also, once you load the program onto a disk for use, you can then put the cassette away in its safe plastic storage box until you foul up the disk copy.

Disks can be zinged by almost anything, while cassettes are virtually indestructible. The plastic box I store them in keeps out the dust and has enough room to quickly index the cassette. In Japan virtually all programs are sold on cassettes.

So what would it cost to have kept the small circuits needed to interface a cassette recorder in the Apple? I appreciate the fact that Apple pioneered the microcomputer floppy disk—but this may have been going a bit too far. Of course, by not permitting further use of the cassette, they did force people to think about disks with a vengeance. But it also wasted a lot of customer money.

Another area given too little shrift by many manufacturers, including Apple—and certainly Radio Shack—is cooperation with computer magazines. There are many of them, at first appearance; but when you remember that few buyers of new desktop computers fail to ask a friend who already has a computer about their purchase, it's clear that virtually every new purchase of a computer depends, to some degree, on how experienced computerists view the system.

We'll see what happens with the Lisa; but unless the world has changed in the last year or two, it is going to be like pulling teeth to get a review of Lisa for our magazines. Yet, we have over one million readers who will probably have more to say about the success of the Lisa than any Apple ad in *Time* magazine or on television.

Let me ask you a question. You have a computer. Is there a week when someone doesn't ask you for a

"Computer magazines are being almost totally ignored by our industry."

recommendation about computers? With me it is an hourly question, not a daily one. Do I recommend a computer I don't know about? Don't be silly. No more than you do. The Fortune computer may be great, but I've never had my hands on one—or talked with anyone who has—nor have I read much about it in computer magazines. So I ignore it when asked about it.

Many of you talk to local groups about computers, drawing on what you've learned from your own computers and on what you've read in computer magazines. Yet computer magazines are being almost totally ignored by our industry. A recent count of firms manufacturing desktop computers was well over 250, yet most material in computer magazines revolves around a dozen systems. Many marketing managers are not doing their jobs well. They don't seem to understand what is going on. ■

In Defense of Gaming

by George M. Engel

What do you do when you get those silly questions, like, "What does it do. . . besides games?" I stifle my anger and launch the doubter into two uninterrupted hours of alf music, adventures, math tutorials, DNA molecules, flight simulators, graphics demos, word processors, VisiCalc, VisiPlot, and so on, ad infinitum. The parting comments are always, "I didn't realize," or, "I thought it was only for games."

ONLY for games? That's why I bought my Apple. I have many other programs, of course, but I'm a Gamer first and foremost. Why be ashamed to admit it? But, let's go back a bit. . . .

Remember when you first learned math and spelling? It was never fun and exciting. Come to think of it, it was darn hard work. Think what you could have done if you'd really enjoyed it—if they had made a game out of those subjects. Today, you put your children in front of your Apple with a good game and look what they learn: spatial relationships, eye-hand coordination, and so on.

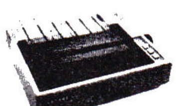
What else did we forget? The most important thing—responsibility. On the computer, children are responsible for their own actions. Their mistakes are their own. They can't blame anyone else. They're not working as a team, they're working as individuals.

Our colleges train students to enter the labor force. "Don't be different," is the message. "Don't find the creative way to do things, you'll embarrass others." Our colleges prepare students to find jobs. European universities train students to make jobs! Let's train our kids as individuals, not as a team; to assume responsibility, not disseminate it; to nurture creativity, not stifle it.

Games like "Cartels and Cutthroats," a business simulation, bring out this wonder called creativity and force you to think as an individual. You have to. It's *your* business. Adventures, strategy simulations and war games all make you think. They stimulate the reasoning process, and make it fun to learn, to create, to assume command and to assume responsibility. That's life, my friends—learning to cope with the world on its own terms.

All we can do for our children is provide the tools and the stimulation to succeed. As Apple owners we have the tools. They *are* expensive, but, as the slogan goes, if you think education is expensive, consider the cost of ignorance. Your kids will have fun learning on the Apple. Encourage and applaud them when necessary. Then they'll take over. We are training them to take over, aren't we? The world belongs to our children and the next decades are theirs. We're only the caretakers.

That's as simple as I can make it. You're remembered in this life by how well you play the game. I hope my thoughts help the next time you're asked, "But what does it do?" As always, I can't wait to be asked—and neither can my sons! ■

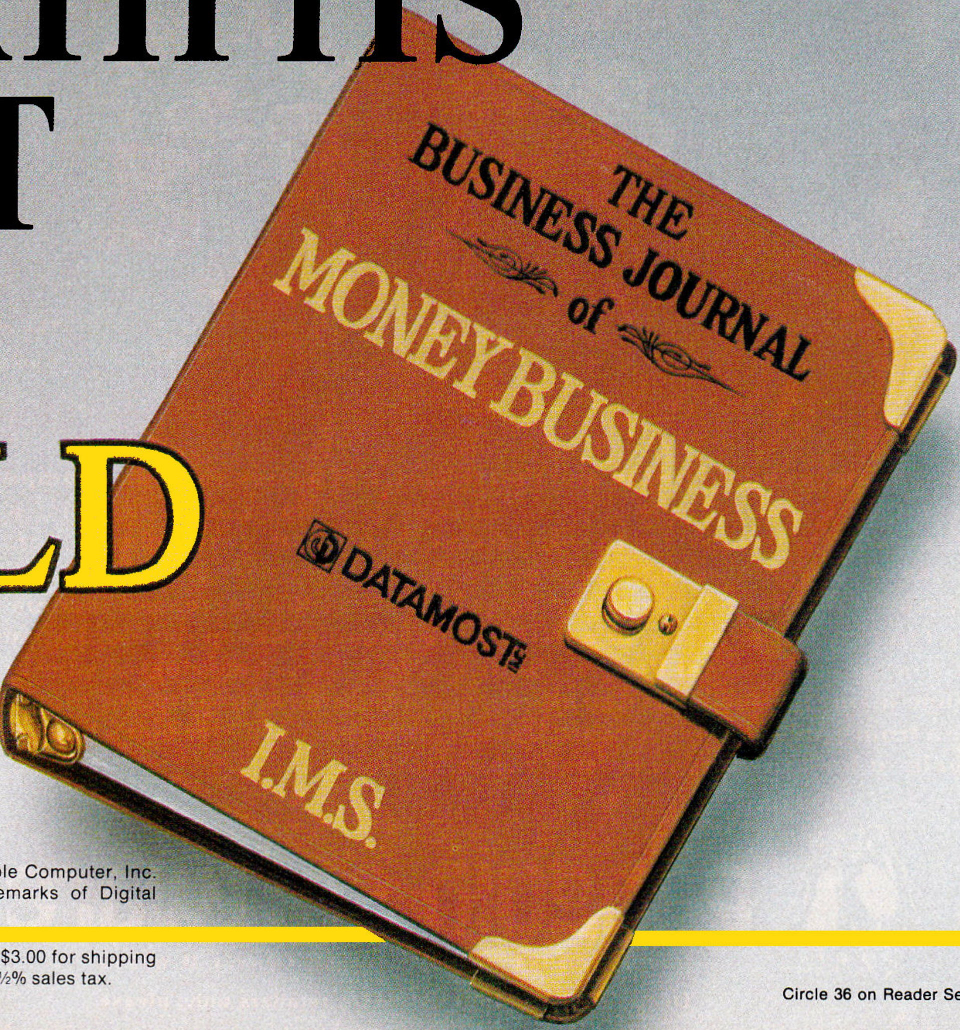




INTRODUCING MONEY BUSINE

I.M.S.

WORTH ITS WAIT IN GOLD



*Apple is a registered trademark of Apple Computer, Inc. CP/M and MP/M are registered trademarks of Digital Research Corp.

VISA/MASTERCARD accepted. Include $3.00 for shipping and handling. California residents add 6½% sales tax.

Circle 36 on Reader Service card.

SS

If you've been waiting and waiting for a truly powerful information management system that's easy to use and affordable, you'll want to get with our program!

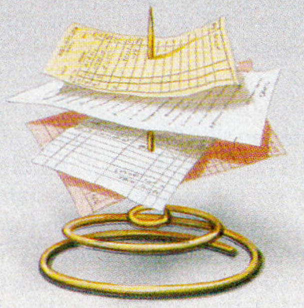
The Wait.
Why did we wait so long to introduce Money Business / I.M.S.? Simple. We'd rather bring you a program a little later, if we can bring you a program that's a whole lot better. And Money Business/I.M.S. is a whole lot better.



While some competitive programs were willing to make compromises in order to make quick sales—we resisted the temptation. We've invested thousands of hours in research and development to make I.M.S. thoroughly professional, reliable, and so easy to use that it doesn't take a computer fanatic to run it. That type of usability takes time to perfect, but it's worth it.

The Gold.
Money Business/I.M.S. won't cost you a King's ransom. At only $495.00 retail, it costs several hundred dollars less than competitive programs which, quite honestly, are neither as easy to use or as powerful as I.M.S.

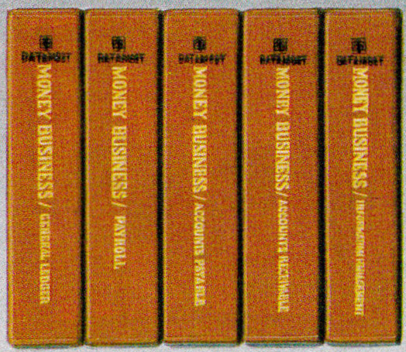
You save money right from the start, since no costly training period or programming expertise is required. So I.M.S. can be up and working in just a few hours, saving you even more money as you enjoy higher productivity through better information management.



The Works.
Exactly what does Money Business/I.M.S. do for you? Briefly, you can create, manipulate and maintain files of data quickly and easily. You can easily create a flow chart of your entire operation in a few hours; and keep it current in just minutes per week.

And, you'll be happy to know that I.M.S. won't force you to change a system you may already be using. For example, you can define screens in any format you desire. Then you select the information your file will contain and how to access it. If you're ever in doubt, just use the help key. I.M.S. contains nearly 100 HELP screens, so the answers are in front of you whenever you need them.

Impressed? There's more. I.M.S. requires absolutely no sorting. It automatically puts all data into place—so you have instant access, saving countless hours of inputting data! I.M.S. can also revise existing files without affecting data that's already entered. It also interfaces with Palantir, Wordstar, Magic Wand and many other popular word processing programs.



The Close.
Money Business/I.M.S. manages your information effectively, efficiently and effortlessly. And it interacts with the entire system of DATAMOST Money Business modules, including General Ledger, Accounts Receivable, Accounts Payable, Payroll and Financier. Six interactive programs in all—tested and proven to perform for you.

Don't Wait. Put Money Business/I.M.S. to work for your business today. You'll be on your way to owning the finest total business program package available.

Money Business/I.M.S. $495.00 at your computer/software dealer, or from:

The Specifications.
Money Business modules run on any computer, including Apple* which uses CP/M* or MP/M*. 48K, at least one disk drive and an optional printer are required to operate the system.

8943 Fullbright Avenue, Chatsworth, CA 91311
(213) 709-1202.

DATAMOST INC

Circle 36 on Reader Service card.

inCider

inCider editorial offices
Pine Street
Peterborough, NH 03458
603-924-9471

What's New This Month?

Welcome to the April issue of *inCider*. In our cover article, Harry White describes several file protection techniques that are easy to implement. A friend of the author, a school teacher, needed a system to deter the mischievous tamperings of students. It seems that a student had left an unusual message for his instructor in an attendance file program. Both hardware and software schemes are discussed, so you can tailor protection to individual needs. Although the protection techniques won't guarantee security, they're sure to deter all but the most determined vandal.

If you're wondering if most of your software will run on the new Apple IIe, read our technical editor's IIe software compatibility study. Over 250 programs, from "A Steller Trek" to "Zork I," have been tested. The study includes some new programs, a few 3:2 DOS favorites, and a number of old but unforgettable classics.

Our guest editor this month, George Engel, isn't afraid to admit that he purchased his Apple mainly for recreational purposes. The author is a Gamer at heart, and he points out that games can develop more than just eye-hand coordination. The value of computer games, especially for the young, is a controversial issue. We'd like to hear how you feel about the issue. If you feel strongly one way or the other after reading Mr. Engle's article, please write down your thoughts and send them to *inCider*.

In the Applesoft Adviser, Dan Bishop continues his tutorials for the Basic programmer. Even an experienced programmer is likely to find something of interest. And for those who are interested in the nuts and bolts of programming, a new column, written by none other than Randy Hyde, author of the LISA compiler, begins this month in *inCider*.

Greg Glau's Bent on Business column describes a novel approach to employer-employee communication. Glau's formula allows an employer to graphically show employees how their performance is helping or hurting the company. It also gives employees a general evaluation of the company's overall performance without revealing actual profit amounts.

Bert Kersey of Beagle Bros, formerly a bored, free-lance graphics designer, tells of his early fascination with computer programming; and in Practically Pascal, John Stephenson tells how to create a useful address book in a language that is not as complicated as is often thought.

This month Paul Raymer introduces a brand new operating system for the Apple, affectionately called CON:DOS. Individual units within the DOS aid in understanding the program and are also quite innovative. Fred Huntington explores the current mis-state of user documentation, and Don Fudge digs deeper into those colorful mysteries known as Apple graphics.

Whether you're a novice or an advanced computerist, we hope you enjoy the April issue of *inCider*. And next month we'll be featuring a program that will help you enjoy the Indianapolis 500 at home. ■

Problems with Subscriptions: Send a description of the problem and your current and/or most recent address to: *inCider*, Subscription Department, PO Box 911, Farmingdale, NY 11737.
Change of Address: Send an old label or a copy of your old address and new address to: *inCider*, PO Box 911, Farmingdale, NY 11737. Please give eight weeks' advance notice.
Microfilm: This publication is available in microform from University Microfilms International. United States address: 300 North Zeeb Road, Dept. P.R., Ann Arbor, MI 48106. Foreign address: 18 Bedford Row, Dept. P.R., London, WC1R4EJ, England.
Dealers: Contact Ginnie Boudrieau, Bulk Sales Manager, *inCider*, Pine Street, Peterborough, NH 03458. 800-343-0728.

Manuscripts are welcome at *inCider*. We will consider publication of any Apple material. Guidelines for budding authors are available. Please send a self-addressed envelope and ask for "How to Write for *inCider*." *inCider* is published monthly by Wayne Green Inc.

Super Stereo Update

I was interested in Phil Roybal's article on phonograph pickup-arm tracking error (January 1983) because the results he quoted matched closely a similar analysis I did several years ago while designing my own arm.

There is, however, an important ambiguity in the article that could result in a number of worried audiophiles and perhaps even some damaged equipment. The important point to add is that *no* manufacturer, to my knowledge, still makes a straight arm for which the equation for underhang can be used. This equation *is* correct, but *only* for an arm in which (1) the axis of the stylus intersects the axis of the arm pivot and in which (2) the position of the arm pivot is fixed.

All of the servo-controlled parallel-tracking arms satisfy condition 1, but in these arms the pivot moves in order to maintain essentially zero tracking error. Because the pivot moves, the term underhang, as defined in the article, has no meaning. Perhaps there were arms in 1941 (when the reference quoted by Roybal was written) that satisfied both conditions 1 and 2. However, *all* modern fixed-pivot arms (even those in cheap toy phonographs) *are* offset.

In other words, all modern fixed-pivot pickup arms should be set up with overhang as calculated from Mr. Roybal's second equation. Straight arms should not be shortened or repositioned to achieve underhang.

R.T. Chilcoat, Ph.D.
State University of New York
Syracuse, NY

DOS Depression Update

A number of readers sent in this simple solution for DOS Depression (Jan:136).

```
D$ = CHR$(13) + CHR$(4)
```

Larry E. Nix
Martin Sandry
Craig E. Daniels
F. Kuechmann

Graphics Tips Wanted

I just purchased your premiere issue of *inCider* and I have thoroughly enjoyed experiencing the beginning of a fresh young magazine. The articles are thorough and easily understood so that implementation of the programs is easy. I enjoyed your Apple/80 article and I look forward to entering the data onto my diskettes and playing the program on my system. Being a graphic designer and a new Apple owner (Apple II) my interest lies in the computer graphics area. I hope your readers (including myself) will contribute new tips and techniques in the graphics field (not just game symbols) and real graphics applications without elaborate hardware and expensive software.

Tom Milligan
San Jose, CA

Printer Help Needed

I have owned an Apple II Plus system for only a few months and have found it to be able to do amazing, complicated things, but I haven't been able to make it do something I consider quite simple. I have the following equipment: Apple II Plus with one Disk II drive, Monitor III, and a NEC PC-8023A dot matrix printer with a Microteck parallel card.

The problem is in introducing printer commands in a copy-protected program. I purchased a PFS software program (Software Publishing Corp.) to do filing, making name and address lists, etc. The program prints nicely but only in single columns. I want to be able to fill the whole page with two or three columns with the listings. For example, if I were printing an address list I would want about six names and addresses at the left margin, six in the center of the page and six on the right side (alphabetized). I have written Software Publishing and they tell me it can't be done. Can anyone help me?

H. Paul Bruns
Royal Oak, MI

edited by John P. Mello, Jr.

HARDWARE

Lisa

Apple introduces state of the art in user friendliness.

With appropriate fanfare, Apple Computer Inc. unveiled its Local Integrated Software Architecture system, known for months as Lisa.

The product of $50 million and 200 man-years of development, Lisa is a software-hardware powerhouse aimed at the office environment.

For a tad under $10,000, the Lisa package includes:

- A Motorola MC68000-based microcomputer with 32-bit internal architecture and 16-bit external datapath;
- Three interface ports, two serial and one parallel;
- A 12-inch, black-on-white, high-resolution (364 lines × 720 dots) monitor;
- One megabyte of RAM;
- Two Apple 871, 5.25-inch disk drives providing 1.7 megabytes of formatted storage;
- ProFile, Apple's 5-megabyte hard disk; and
- Software for word processing, spreadsheet analysis, graphics, graphic design, database management, and project management.

And if all that didn't make Lisa "sexy" enough, it's designed to reverse the slave-computer relationship many users have had with their *Wundermaschine*.

"Lisa embodies a radical change in how users work with computers," the vice president and general manager of Apple's personal office division, John Couch, said in a statement. "Conventional computers created obstacles for those who wanted to make their jobs more efficient. We used progress in microtechnology, plus advances in software technology, to remove many of those obstacles and to make a computer that really is simple to use."

According to the statement from Apple, Lisa's screen shows simple pictures of documents, folders, and other familiar objects in a typical office. A palm-sized device, called a mouse, is used to point to and manipulate the items.

The user controls Lisa intuitively by pointing at and selecting the symbol for file folder, memo pad, waste-



basket, or other familiar objects.

Once the object is selected, it's used like its real-life counterpart. A file folder can be opened and the contents revealed; documents can be refiled, copied, taken out and changed, put into a new file, or thrown away.

The basic functions in all six of Lisa's integrated software packages operate in the same way.

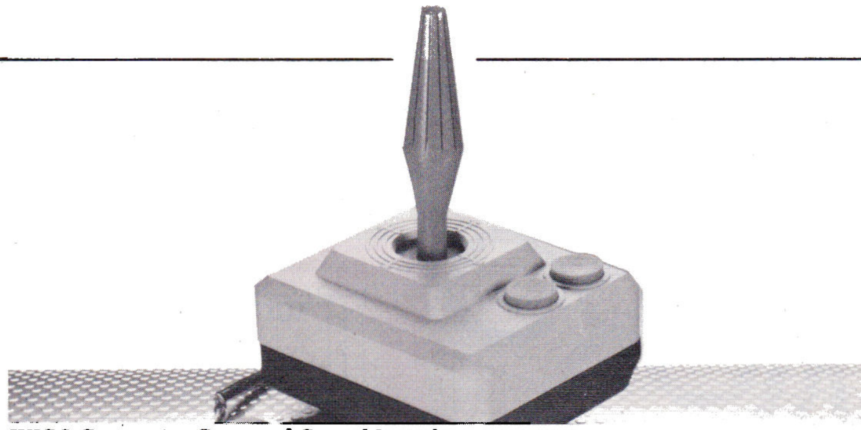
Transferring information from one application to another is easy with Lisa, Apple said. When writing a cost analysis report, a user can quickly shift from word processing to spreadsheet analysis. He can even "cut out" the spreadsheet and "paste" it into his report.

The software packages with Lisa include:

- LisaCalc, a spreadsheet program for handling budgets, forecasts, and other row-and-column models;
- LisaWrite, a word processing program, including a feature allowing users to choose bold, italic, and underlined text via the mouse;
- LisaGraph, a graph display program allowing spreadsheet data to be displayed as bar, line, or mixed graphs;
- LisaDraw, a graphics program allowing users to enhance reports by drawing visuals for them;
- LisaProject, a time-management program permitting a user to keep track of complex deadlines; and
- LisaList, a database management program.

By allowing Lisa to support the CP/M and Xenix operating systems, and run Basic, Pascal, and Cobol, Apple hopes to encourage independent software vendors to write programs for Lisa. Late this year, Apple will also be providing independents with a Lisa Applications Development Toolkit to help them develop integrated programs for the new machine.

Lisa and a communications package for it, LisaTerminal, are expected to be available this spring. Foreign language versions of Lisa will be introduced worldwide this summer. And networking packages for Lisa will be available late this year. ■



WICO Corporation Command Control Joystick
Modified grip handle and two firing buttons.

BUSINESS

Winter Consumer Electronics Show

For Apple users enough new products to crash the senses.

By Robert G. Fisher
Special to inCider

By any yardstick, the 1983 Winter Consumer Electronics Show in Las Vegas was a rip-roaring success. And, for anyone involved with the distribution, marketing, retailing or use of Apple microcomputer products, it was overwhelming.

Officially, 78,126 persons visited the 1050 exhibition booths during the four-day show, breaking all attendance records both for the semiannual show and for Las Vegas, which considers itself the convention capital of the world.

Actually, the number of visitors was far in excess of the official tally; indeed, it may very well have been half again that many. Convention officials explained that local businesses and guests were not included in the official count. And, because WCES passes were limited, daily show access badges were freely passed around by some offical delegates to their friends and "groupies" who wanted to find out what the future of the exploding world of electronics has to offer.

The vast number of products for Apple systems was exhausting to many attendees. On display were a myriad of innovations and improvements in areas of hardware, software, peripherals, furniture and accessories.

By far, the largest group of software exhibitors at WCES were the producers of home video games, many Apple compatible.

Datasoft Inc. had one of the largest such displays, again showing its expertise in adapting its software not only to the Apple, but to a dozen other microcomputer systems as well.

Similarly, Datamost Inc., formed in early 1982, had a giant display, introducing five new computer game cartridges. Datamost expects to log $10 million in software sales this year.

Scholastic Inc. also debuted a new line of children's software, scheduled to be available to the public in the spring under the Wizware label, the latest addition to a full line of that firm's children's communication products. Six program titles, compatible with Apple II Plus, are being made available this spring, including *Microzine*, the first computer maga-

zine for children, and other fun video learning games.

In the peripheral area, Star Micronics Inc. of Dallas introduced a new thermal printer, priced to retail for $199, that promises to pierce that market.

Brother International displayed its new electronic compact typewriter line, the Correctronic 50, capable of doubling as a microcomputer printer terminal by use of a built-in interface port.

Discwasher of New York introduced a disk drive cleaner, designed to remove all debris and keep disk drives working at optimum capacity. It is non-abrasive and utilizes a unique fiber grid cleaning system and "perfect path" technology to thoroughly dislodge and collect foreign matter from sensitive drives. The cleaner is available in 5¼-inch and 8-inch formats.

A new WICO Corporation Command Control joystick, first seen at the show and interfacing with the Apple II and other personal computers, is being made available. Equipped with a modified grip handle and two firing buttons, plus a selection switch for centering/non-centering option, it will retail through Apple outlets at $69.95.

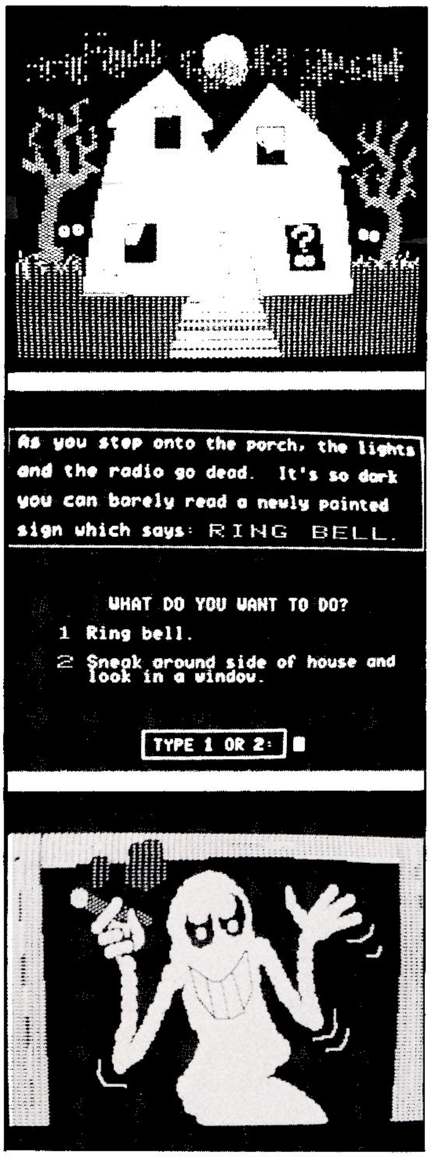
Both Maxell and BASF and other tape manufacturers competed for attention at the show for their respective segments of the tape and floppy disk market.

BASF exhibited its Qualimetric FlexyDisks, first introduced in October 1982, and claimed to be the first manufacturer of 100-percent, error-free floppy disks with a lifetime warranty.

Maxell Corporation delegates proudly demonstrated their revolutionary 3-inch Compact Floppy Disk (CFD), along with their compatible (and same recording capacity) 5¼-inch double density, mini-floppy disk.

At a computer seminar during the show, leading industry spokesmen noted that, while computer and computer-related sales were up over the preceding year generally, those sales still were not as much as originally projected, due in great part, they said, to the faltering economic situation.

Nevertheless, most executives foresee an economic turnaround that will benefit the personal computer industry and its spinoffs. Even as the WCES was drawing to a close, those industry representatives were planning for what they expect to be an even bigger and better show next June in Chicago. ■



Scholastic Inc. of New York
Six new programs compatible with the Apple II Plus.

HARDWARE

William Tell Overture

A new Taiwanese micro sporting a 6502 and Z-80 intends to take on Apple.

By Robert Fisher
Special to inCider

A new microcomputer is being introduced into select retail areas this spring that—if for no other reason than its price tag—promises to make a solid impact into the personal computer market.

Aptly named the William Tell, manufactured by Star Micronics Inc. of Dallas, TX, it will retail for an extremely competitive $999.95. It was briefly shown at the 1983 Winter Consumer Electronics Show in January and will make a repeat appearance at the June CES in Chicago.

David Wooldridge, consultant to Star Micronics, describes the William Tell as "basically, a 64K system, CPM-compatible, with a Z-80 microprocessor. And, so it will be compatible with Apple systems, it also carries a 6502 processor.

"We're carrying both sides of the street," Wooldridge said. He pointed out the new microcomputer also has a universal power supply (110V-220V, 50- or 60-cycle) and can be operated anywhere in the world.

Additionally, Wooldridge explained, it has a full color output for monitors and is equipped with full graphics. And, for those who want a personal computer providing home entertainment, the William Tell has

a port to plug in a joystick for video game playing.

Asked whether the product name, William Tell, had any significance, Wooldridge smiled and said, "We imply that another product is a target, but we never come right out and say so."

The William Tell is manufactured with two 5¼-inch disk drives and has a 10-key pad. Upper- and lowercase character sets are built into the system, and, because it is CP/M compatible, it will make an 80-column printout. "With the William Tell, Star Micronics is putting it all on one card," Wooldridge said. "It is especially suitable for printout operation with Star Micronics' Gemini 10 and Gemini 16 printers, as well as Star's new thermal printer that retails for $199," he added.

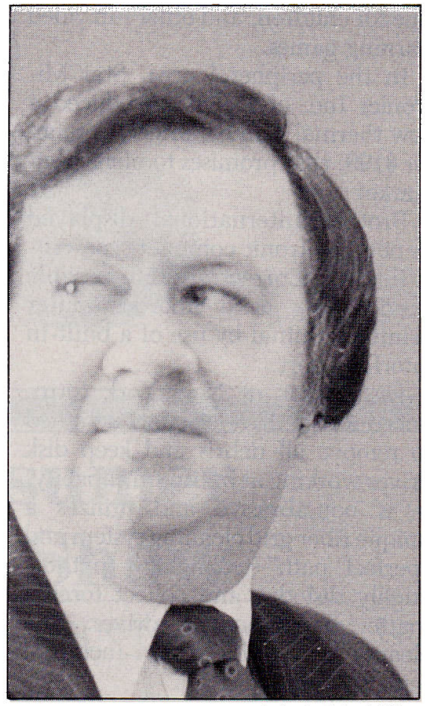
"It's an excellent system that can be taken home for a very inexpensive amount of money," Wooldridge said. "For about $1500, a user can get the William Tell, a good monitor and a printer."

For those who do not want a disk drive, the William Tell also has cassette recorder connection terminals.

Marketing of the William Tell is based in Dallas. "We're trying to select some of the distributors with whom I've been doing data processing business for some 20 years—distributors who have proven track records," said Wooldridge, a former National Cash Register executive.

"We're getting the distributors that really reach the market, and we're giving some points that will let them and the retailers make some money, without relying totally on followup sales of peripherals and software.

"It's like a crap table," said Wooldridge during a CES interview in Las Vegas, the gaming capital of the world. "You can't bet on a come. And that's what you're betting on when you rely on selling software to make a profit. If you can't make the subsequent sales, there's no way you can justify the floor space, warehousing, salaries and overhead for a few dollars a unit.



photos by Robert Fisher

"For about $1500, a user can get the William Tell, a good monitor and a printer."

David Wooldridge, consultant to Star Micronics of Dallas

"We imply that another product is a target, but we never come right out and say so."



"We're doing our best to pick distributors that will maintain our price structure. Of course, that cannot be guaranteed, and we're not so naive as to think that.

"We're picking distributors that get as close to retail as they possibly can, but people who are not discounters. There's not anything wrong with discounting, except a tendency to have a problem one way or another; discounting drives the retailers down."

At the June CES in Chicago, Star Micronics intends to have still another new product: a fully portable, disk-driven microcomputer with a 9-inch green-phosphor screen that will also carry a 6502 Apple-compatible board.

Prototypes of the portable unit were produced in January, and the firm approved the design of the case at that time. Initial distribution will begin in May, and dealers will see it in full bloom at Chicago's CES, Wooldridge said. ■

BUY SOME TIME FOR $169

Time is money. You have a computer system because you know that it saves you money by simplifying procedures and reducing time normally involved in your work. Time is an important resource which should not be wasted. You are wasting valuable time if you ever wait for your printer.

No waiting. Now with Microfazer by Quadram there is no more waiting.

Microfazers are inexpensive universal printer buffers which any computer user cannot afford to be without. Any computer—any printer (or plotter!), whether parallel or serial. Microfazer receives information from the computer, at ultra high speeds causing the computer to think the printer is printing just as fast as the computer can send. Microfazer holds the information until your printer can handle it, and then sends it on.

More copies. Microfazer is equipped with a copy feature allowing additional copies of the buffered information—from one to as many as you want—with the mere press of a button. When you need your information repeated, for whatever reason, it's always right there—inside Microfazer.

Microfazer™ stack. Microfazer can be stacked with popular modems or other peripherals. Some models can plug directly onto the back of your printer. Install it in less than 60 seconds, and choose the amount of buffer memory you need—8K, 16K, 32K, or 64K. One model even comes with up to 512K! You may use several Microfazers in series to create just what's right for you. Take data in from a serial computer and out to a parallel printer. Or in from a parallel computer and out to a serial printer. Microfazer is just as flexible as you need it to be.

Low price. Only $169 for 8K of buffering, $189 (16K), $225 (32K) and $299 for a full 64K. Serial-to-Parallel, Parallel-to-Serial and Serial-to-Serial models have slightly higher prices.

QUADRAM CORPORATION

4357 Park Drive / Norcross, Ga. 30093
(404) 923-6666



Special version for attaching directly to the popular Epson printer.

Circle 234 on Reader Service card.

by Dan Bishop

Conditional Relationships and Subroutines

In the previous article in this tutorial series covering the use of Basic, I dealt with the principal looping structure provided by Basic, the For/Next loop. Data input using the Read command in combination with Data statements was also introduced, along with the Peek/Poke commands and how these two commands could be used to allow the programmer the flexibility of restoring the data pointer to any desired position in the data list without having to always restore to the beginning of the data list.

In this article, I will deal with the If/Then statement and the use of subroutines. I hope that even the veteran Basic programmer will spend some time with the material in this article, for I anticipate that some of the concepts that I will present may not be widely known or used. I will introduce the concept of structured programming and show how the If/Then/Else, and While/Do, the Repeat/Until, and the Case/Of structures found in such languages as Pascal can be duplicated in Applesoft Basic.

Conditional Branching In Basic—If/Then

Without the capability to make decisions based on elementary relationships and logic, the computer would be a very ineffective tool. As it is, the computer can make simple comparisons and use the results of such comparisons to branch to different sets of instructions. Although the result of any comparison must be either "true" or "false" in a strictly logical sense, the flexibility this gives the computer is awesome.

Numbers or mathematical expressions can be compared for equality, and the result will be either "true," meaning "yes, these two numbers (variables, expressions) are equal" or "false," meaning "no, these two are not equal." Variables are most commonly used in these comparisons, and it is, of course, the values that these variables represent that are being used by the computer for the comparison.

The principal Basic instruction that directs the computer to make such a comparison is the If/Then statement. The general format for this instruction uses one of the following forms illustrated in Table 1.

```
IF expression 1 =  expression 2 THEN...
IF expression 1 <> expression 2 THEN...
IF expression 1 <  expression 2 THEN...
IF expression 1<=  expression 2 THEN...
IF expression 1 >  expression 2 THEN...
IF expression 1>=  expression 2 THEN...
```

Table 1. If/Then forms.

In the statements shown in the table, the symbols separating the two expressions are referred to as "relational operators." In the order shown, they are read as follows: "equals," "does not equal," "is less than," "is less than or equal to," "is greater than," and "is greater than or equal to." Expression 1 and expression 2 may be simple variables or constants, as in the following:

```
IF A=45 THEN...
IF XY> A3 THEN...
```

The ellipses following the word THEN represent any valid Basic command or combination of commands separated by colons. They represent the specific set of instructions that the computer is to follow should the relationship between the two expressions turn out to be true. If the relationship is not true, the computer will completely ignore all of the commands contained in that program line and will continue with the next program line. For example:

```
60 IF A=45 THEN PRINT "A EQUALS 45.":GOTO 80
70 PRINT "A DOES NOT EQUAL 45"
80 program continues from here.
```

Notice that two commands are included following THEN in line 60. These commands are only carried out if expression 1, the value associated with the variable A, equals expression 2, the constant 45. If A is not equal to 45, the computer will simply ignore these two commands and go on to line 70. The Goto command in line 60 is very important. Without it, for the case that A does equal 45, we would see the following displayed on the screen:

```
120 IF AR >=BC THEN GOTO 220
130 beginning of instruction block for "false"
    ...
    ...
200 end of instruction block for "false"
210 GOTO 470
220 beginning of instruction block for "true"
    ...
    ...
460 end of instruction block for "true"
470 program continues from here
```

Sample listing 1.

Address correspondence to Dan Bishop, Custom Comp, PO Box 429, Buena Vista, CO 81211.


THE INCREDIBLE
JACK.
T.M.
for the
IIe
PERSONAL
FILING
WORD
PROCESS
CALC
MAILING
LABELS

A EQUALS 45.
A DOES NOT EQUAL 45.

The GOTO command forces a jump over the instruction block that deals specifically with the situation occurring when the relationship is found to be false.

This kind of approach begins to take on a leapfrog appearance when the instruction sets for the true and false situations become more complicated. For example, the situation in Sample listing 1 uses lines 130 to 210 to handle the instruction set to be carried out if the relationship between the two expressions is false, and lines 220 to 460 if the relationship is true.

This leapfrogging, although encouraged in Basic, brings nervous tremors to programmers who adhere rigidly to "structured programming" concepts, and we shall see later how to avoid this problem. But first let's finish with the If/Then statement as it is interpreted by Applesoft Basic.

String expressions may be compared, but you must remember that the strings are compared literally, character by character, throughout their entire length. The comparison is based on the ASCII code for each character being compared, and a shorter string that is identical to the first portion of a longer string is taken to come before, or be "less than," the longer string. The following comparisons would evaluate as "true":

```
IF "APPLE" < "PEACH" THEN...
IF "APPLE" > " APPLE" THEN...
IF "DOG" < "DOGWOOD" THEN...
```

In the second example, expression 2 begins with the space (ASCII code 32), while expression 1 begins with an A (ASCII code 65). Although these examples use string constants, string variables could also be used in the comparisons.

The literal evaluation of the relationship between the two expressions can cause problems if the expressions represent real numbers that result from calculations. If the two expressions evaluate to two real numbers that are identical except in the last decimal place retained by the computer, they will be considered unequal. This usually is not what the programmer desires. In such cases, it is best to evaluate the expressions as integers that contain the desired number of decimal places. For example, suppose that AB is to be compared with AM, both being real numbers that have been defined earlier in the program. If we desire the comparison to be carried out to the third decimal place, the following expression could be used:

```
IF (INT(AB * 1000 + .5)) = (INT(AM * 1000 + .5)) THEN...
```

In this case, the two real numbers are multiplied by 1000 to accommodate the third decimal place. Each has 0.5 added to it, which rounds the number based on the fourth decimal place, and the resulting numbers are then converted to integers before the comparison. Of course, this approach will only work when the resulting integers are less than 32767, the maximum allowed value for integers.

Finally, two other forms of the If/Then statement are equally valid in Applesoft Basic. When the only instruction following the word THEN is a Goto command, the word THEN may be left out entirely, or the Goto may be left out. The following three statements are equally valid and perform the same function.

```
IF RF > TD THEN GOTO 120
IF RF > TD GOTO 120
IF RF > TD THEN 120
```

Logical Operators

The operators introduced in the previous section are called "relational operators" because the comparisons which use them are based on how the two expressions are related to each other. Applesoft also allows for use of three "logical operators," AND, OR and NOT. Most commonly, these operators are used to compare the results of two or more relational expressions. As used here, a relational expression is simply the relational comparison of two arithmetic or string expressions exactly like the examples presented above.

But when we want the computer to carry out certain instructions only if two such relational expressions are evaluated as "true," our program can be simplified using the logical operator, AND. Several logical relationships are illustrated below.

```
IF (A = B) AND (A>D) THEN...
IF (M$ = "CITY") OR (S$ = "STATE") THEN...
IF (G% = 4) NOT (F% = 7) THEN...
```

In the first of these examples, the instructions following the THEN (as represented by the ellipses) will be carried out only if both relational expressions (A = B and A greater than D) are true. In the second example, the instruction set represented by the ellipses will be carried out if either or both of the relational expressions are true; in the third example, those instructions will be followed if G% equals 4 and F% is not equal to 7.

Obviously, the same results could be accomplished using two or more If statements that use only relational expressions. However, as such relationships become more and more complicated, appropriate use of the logic operators will help keep the program more understandable and easier to work with. There is no practical limit as to the degree of complexity of these logical relationships, either, and they may be nested with the use of parentheses. For example:

```
IF (A = 5 AND B = 7) OR (A = 6 AND B = 5 NOT (D = 3 OR D = 4)) THEN...
```

In this case, the instructions following THEN will be carried out either when A and B are 5 and 7, respectively, or when A and B are 6 and 5, respectively, so long as D does not equal 3 or 4.

Subroutines

A subroutine is a set, or block, of Basic instructions that may be thought of as being a "mini-program" that is separate from the "main program" but that is used by the main program whenever its instruction block is needed. The appearance of the subroutine is no different from that of the main program. Each instruction line has its own line number, just as in the main program. The only distinguishing characteristic is that the last command in each subroutine must be the single word, RETURN.

Subroutines may appear anywhere within a program. However, if they



Extras can more than double the price of your personal computer. Not so with the Franklin ACE 1200. It's the professional computer system that includes the extras—and a long list of exclusive Franklin features that make it the most extraordinary value on the market today.

The ACE 1200 has everything you'll need to add a color or black and white monitor, modem, printer, back-up disk drive and other accessories. You can choose from the enormous selection of Apple programs and peripherals because the ACE 1200 is hardware- and software-compatible with the Apple II. And, with the built-in CP/M card, you can run both Apple II and CP/M programs. Franklin's CP/M operates three times as fast as many competing systems, drastically reducing processing time for most business applications.

The Franklin ACE 1200—the most extraordinary value on the market today. Call or write today for the name of your local authorized Franklin dealer.

Franklin ACE is a trademark of Franklin Computer Corporation.
Apple is a registered trademark of Apple Computer Inc.
CP/M is a registered trademark of Digital Research Inc.
VisiCalc is a registered trademark of Visi Corp.

FRANKLIN
COMPUTER CORPORATION

7030 Colonial Highway, Pennsauken, NJ 08109 609-488-1700

Circle 184 on Reader Service card.

beep

IN THE

are placed within the body of the main program, and are not also being used as part of the main program, a GOTO ##### must be placed immediately before the first line of the subroutine. This causes the main program to jump over the subroutine. The line number used in the Goto command corresponds to the line that follows the subroutine's Return command.

The subroutine itself is called by the main program whenever the block of instructions that it contains is needed. As soon as the computer encounters GOSUB #####, it stops the line-by-line sequential execution of program instructions, sets a pointer to the command immediately following the Gosub statement, and proceeds to the line number specified by the Gosub command. This line number is, of course, the first line in the desired subroutine.

Line-by-line sequential execution of the subroutine's instruction set is then carried out until the computer encounters the Return command. The computer then jumps back to the main program to where the pointer was set and begins executing the main program lines again.

Subroutines may be nested several layers deep. That is, the main program may call a subroutine which itself uses a subroutine, etc. In each case, the computer sets a marker to the statement following the Gosub command and jumps to the specified subroutine. Each Return encountered sends the computer back to the appropriate earlier subroutine, until the last Return sends it finally back to the main program.

There is a limit to the degree to which subroutines may be nested. The computer has only a certain amount of memory reserved for the Return pointers. In the case of Applesoft, this limit is 25. Should the limit be exceeded, the computer will respond with an Out of Memory Error. In this case there may be plenty of memory left for program instructions and variable storage; the Out of Memory refers to the "stack" space for the Gosub/Return pointers.

Problems arise if you are executing a subroutine and decide to abnormally jump out of the subroutine with a Goto command. This will leave a Gosub/Return pointer on the stack throughout the rest of your program, and the computer will always think it is one level deeper into nested subroutines than you actually want it to be. Applesoft provides a way out of this with the POP command. This single word command should be used just before the Goto is executed. An example of how this might be used is shown in Example 1. POP removes the last pointer address from the Gosub/Return stack just as though a Return command had been executed.

```
1150  subroutine begins here
      ...
      ...
1280  IF R% > S% THEN POP: GOTO 350
      ...
      ...
1340  RETURN
```

Example 1. POP command illustrated.

For program clarity, however, such abnormal jumps out of subroutines should be used very sparingly.

Structuring Programs Using Subroutines

There are two main reasons for using subroutines. The first is to avoid having to repeat the same block of code several times within a single program. A specific subroutine can be called up several times by the main program each time it is needed. The Return command will always send the program back to the appropriate spot in the main program from which sequential execution again begins. Subroutines used for this purpose are best placed near the beginning of a program (low line numbers). This improves program efficiency, since Basic begins looking for the Gosub line number at the first line of the program, and continues searching through the line numbers until it finds the one called for.

I generally begin my main program routines at line 1000, reserving the first lines of the program for Remarks and a GOTO 1000 command. This leaves plenty of room for these frequently used subroutines.

A second use for subroutines is often overlooked by the Basic programmer, probably because it is too easy to simply sit down and begin writing code. Programming ought to begin with a carefully thought out design that depicts each of the program's main functions. Then each of these functions should be further defined in terms of sub-functions. Each sub-function should be dealt with in the same way, until the entire program is displayed as a clearly defined sequential execution of fairly simple functions.

Only then should the first line of code be written. Each simple function defined at the end of the program analysis stage may be written as a unique subroutine. The main program then becomes a series of Gosub commands. If copious use of Remark statements is made, the program becomes almost self-documenting. The flow of logic is easily read from the sequence of subroutines. I find it useful to use the line number immediately preceding the first line in a subroutine for an appropriate Remark statement that identifies the subroutine that follows. For example, if my subroutine starts at line 800, then line 799 will have the identifying Remark

```
UNTIL A = B DO...
100  IF A = B THEN 190
110  program block to be repeated
     ...
     ...
180  GOTO 100
190  program continues from here
```

Example 2. Until/Do shortcut.

statement. In this way, if I find that I must remove Remarks in order to make more room for my program, I can find them easily and remove them without fear of deleting a line number referenced by a Goto or Gosub command.

There are several advantages to designing a program in this fashion besides the ease with which it can be read. Once you have become used to writing programs using structured techniques such as this, you will find that the time spent writing a given program will be considerably shortened. You will identify subroutines you have used before that can be simply copied into your new program with little or no change. You will find that new subroutines can be tested out by themselves to verify that they are written correctly, without having to run through the entire program to test a single function. You will find that later, when you come back to the program to make modifications in its operation, those changes will be much easier to make with less chance of interfering with another part of your program.

Conditional Short-Cuts

Basic programs in which the following sequence of instructions (or its equivalent) is used are all too common.

```
IF A > 15 THEN GOTO 820
IF A <= 15 THEN GOTO 750
```

Based on the previous discussion, it is obvious that the second relational comparison is not needed. It merely slows the execution of the program. The instruction GOTO 750 could be used in its place, with the same results.

Flags are often employed in programs to be used in conditional expressions. For example, a certain part of a program might be used for both data entry and for editing. Suppose the common instruction block is part of the main program for data entry, and a flag, say F2%, is zero during data entry. Now suppose the same instruction block is used as a subroutine for editing, with F2% set to 1. The following line could be used to return from the instruction block for the editing routine:

```
IF F2% THEN RETURN
```

F2%, by itself, is taken as a logical relationship and is evaluated as "true" or "false" based on its value. So long as its value is not zero, it will be evaluated as "true."

Two programming structures not supported by Applesoft are the Until/Do and the Repeat/Until com-

```
REPEAT...UNTIL A=B
100  beginning of program block to be repeated
     ...
     ...
180  IF A<>B THEN 100
190  program continues from here
```

Example 3. Repeat/Until command.

mands. These two commands are common in structured languages such as Pascal. They identify a block of instructions that are to be carried out repeatedly until a specific condition is met. The main difference between the two is that the condition is tested before the instruction block is carried out with the Until/Do structure, while the condition is tested after the instruction block with the Repeat/Until command. The temptation in Basic is to set up a For/Next loop and write a conditional test within the loop that will cause the computer to jump out of the loop when the condition is met. This involves another Goto jump, and presents other hazards which I brought out in my March article.

In Basic, the Until/Do command can be accomplished by stating the condition before the loop using an If/Then as in Example 2.

The Repeat/Until is carried out by reversing the conditional relationship and using the If/Then after the block of instructions to be repeated if the condition is not met, as in Example 3. Although both of these constructions in Basic use the Goto command, the jumps are to points immediately contained by the looping structure and present no problems in obscuring the goal of the program.

In fact, if the program block that is to be repeated is itself a complete subroutine, the Until/Do and Repeat/Until constructions become "one-liners":

```
100  IF A<>B THEN GOSUB 1200:
     GOTO 100
```

and:

```
100  GOSUB 1200: IF A<>B THEN 100
```

This approach really tightens up the program logic! Take note of the reversal in the conditional expression for the program line that mimics the Until/Do command.

Emulating the If/Then/Else In Applesoft

Relational and logical expressions are evaluated to +1 if true, and to 0 if false. Knowing this makes it possible to set up the equivalent of an If/Then/Else statement in Applesoft, and at the same time to structure the program so that the leapfrog effect mentioned earlier is entirely avoided. Many advocates of structured programming decry the use of the Goto command, maintaining that the

```
CASE expression OF
    1: command list 1;
    2: command list 2;
       ...
       ...
    8: command list 8
END;
```

Example 4. Case/Of command.

main program logic should flow in an uninterrupted sequence from start to finish with no sudden jumps initiated by a GOTO #####. The If/Then/Else construction, used with subroutines, can eliminate a lot of these Gotos.

The key here is to use the ON. . . GOSUB ####, #### command. When this command is encountered, the computer evaluates the expression (which may be a single variable, an arithmetic expression, or a relational or logical expression).

If the expression evaluates to 1, then the first set of line numbers after the Gosub is used. If it evaluates to 2, the second set is used. In fact, any number of line numbers could be listed in sequence following the Gosub command to correspond to any small positive integer to which the expression could evaluate. If the expression evaluates to the number 8, the computer would look for the eighth line number in the list and execute a Gosub to that subroutine. If the expression evaluates to some number too large to match with a line number in the list, the computer ignores the statement and continues on to the next command. (Incidentally, the same command can be used with Goto instead of Gosub, but this defeats our efforts at structured programming, right?)

Now suppose that we set up two subroutines for a particular conditional test within our program. The first, beginning at line 1500, will contain the instruction set to be followed if our conditional relationship proves to be true, while the second, beginning at line 1600, deals with the false relationship. If the relationship being tested is equality between the two variables, AB and AC, then the following statement could be used:

```
80 ON (AB = AC) + 1 GOSUB 1600, 1500
```

In line 80, the relationship between AB and AC is tested. If the two are equal, the result is +1, to which 1 is added, making the entire relationship equal to 2. Thus GOSUB 1500 will be executed. On the other hand, if AB and AC are not equal, then the result of the relationship is 0. When 1 is added to this result, the GOSUB 1600 command is executed. This is equivalent to:

```
80 IF AB = AC THEN GOSUB 1500 ELSE
   GOSUB 1600
```

Of course, Applesoft does not support the Else statement, so the only way to handle this situation is with the ON. . . GOSUB command. Notice that the main program logic flows smoothly from line 80 to line 90 with no leapfrogging or ugly jumps.

Another construction found in Pascal and structured languages is the Case/Of command. This command is followed by a list of instructions, only one of which is to be selected based on evaluation of the expression in the Case command. This command looks something like Example 4.

If you are ahead of me, you may recognize the similarity between this and the ON. . . GOSUB construction that we have already dealt with. In the Case/Of construction, if the expression evaluates to 1, then the first command list is carried out; if it evaluates to 2, the second command list is used; and so on. In Basic, if each command list is made into a separate subroutine, then ON. . . GOSUB carries out the same function.

The relationship does not stop here, however. The Case/Of command can also use letters or complete words to specify which of the command lists to carry out. Basic also has this capability, if we are ingenious enough to construct the proper relational expressions to use for our ON. . . GOSUB expression.

Consider the Case/Of expression in Example 5, which calculates the volume of various solids: The Pascal program will evaluate the string variable named SHAPE, and carry out the desired calculation if SHAPE is equal to "cube," "box," or "sphere."

```
CASE SHAPE OF
  CUBE:     VOL = S * S * S;
  BOX:      VOL = L * W * H;
  SPHERE:   VOL = (4/3) * PI * R * R * R
END;
```

Example 5. Case/Of expression.

In Basic, if SH$ represents the shape of the object, then we can formulate relational expressions to use in an ON. . . GOSUB statement. For example, the expression (SH$ = "CUBE") will evaluate to +1 if true and to 0 if false. If we multiply this expression by some number, then the result will be that number when the expression is true, and zero when it is false. The Basic construction that carries out the Case/Of example shown above then becomes:

```
ON (SH$ = "CUBE") * 1 + (SH$ = "BOX") * 2
+ (SH$ = "SPHERE") * 3 GOSUB 1100, 1150,
1200
```

So if SH$ is "SPHERE," the entire expression becomes 0 + 0 + 3 = 3 and the third subroutine, at line 1200, will be the one selected.

Conclusion

Although no specific applications programs were given in this article, I hope that you will find numerous applications for some of the concepts that I have described. Take a look at some of the programs you have written already. Count the number of Goto jumps within the program. Then see if you can redesign the program with an emphasis on individual functions. Write subroutines for these functions and then put the program back together using these modules and eliminating as many of the Gotos as you can. Not only can this be an interesting exercise, but it can help you begin to think in terms of structured programming. You may never be the same again!

In my next article, I will modularize random access disk file routines and suggest how it may be possible to use the disk for random access data storage with very little confusion or headache. If you have been avoiding random access files because sequential files were more familiar, then the time has come to graduate to the faster, more versatile world of random access. You will be surprised at how easy it can be! ■

by Randall Hyde

Trying Out the Tools—Some Basics

Once the beginner masters his Apple computer and develops a strong command of the Basic language, thoughts immediately turn to optimization. There's probably not a single program written that couldn't benefit from extra speed or additional features. So the beginner learns Basic tricks, like removing Rems, moving subroutines to the beginning of the program, and declaring often-used variables early in the program.

Ultimately, however, jerry-rigging fails to achieve the needed improvement and the programmer is forced to contemplate use of a faster language. While faster high-level languages, like Pascal and Forth, are available for the Apple II, these languages can't come close to 6502 assembly language. For the programmer writing time-critical code (like a hi-res arcade game) using 6502 machine code is an absolute necessity.

However, 6502 assembly language, the gateway to 6502 machine code, is difficult to learn. To ease the burden I have created a set of subroutines, collectively called Speed/Asm, to help minimize the effort. This series of articles will describe how you can easily create your own machine language programs using the Speed/Asm package.

Speed/Asm, like its companion, the LISA interactive assembler, is especially designed for the beginner at 6502 assembly language programming. I developed these packages as tools for use in the assembly language classes I teach in southern California. I've discovered that students who learn 6502 assembly language using the Speed/Asm package achieve competence much faster than those who learn using traditional methods of instruction.

Speed/Asm is a collection of subroutines you call from your program to perform certain tasks. In particular, the Speed/Asm subroutines emulate many of the statements found in Basic and Pascal. For example, the Basic program

```
10 FOR I = 1 TO 100
20 PRINT I;
30 NEXT I
```

is translated into Speed/Asm as

```
JSR FOR
ADR I,1,100
JSR PRTINT
ADR I
JSR NEXT
```

Persons using a macro assembler, like LISA version 3.0, could even code this as

```
.FOR I,1,100
.PRTINT I
.NEXT
```

As you can see, Speed/Asm looks quite a bit like Basic. But keep in mind that Speed/Asm *is* assembly language so you get a considerable performance boost compared to Basic.

Getting Started

Probably the best place to start is with the disk that comes in the Speed/Asm package. A quick catalog of the disk reveals that about 11 programs are included.

The file Speed/Asm is the boot program that does the cataloging. RELSA and RELFP are two programs used to generate a Speed/Asm program. Speed/Asm is relocatable to any page boundary; RELSA and RELFP are the programs that create an absolute version of the Speed/Asm program for actual use. RELFP generates a full Speed/Asm package, including the floating point operations. RELSA generates a copy of Speed/Asm without the floating point subroutines, hence it is much shorter.

SAFP.78, the next program on the disk, is a copy of Speed/Asm that has been located to address $7800. This file is provided for the convenience of those who want to write programs as quickly as possible without having to learn how to use the RELSA or RELFP program first. I will use SAFP.78 in all the examples I present, although only one line in your program will need to be changed if you wish to use Speed/Asm located at some other address.

The SA.EQUATES file contains equates for all the Speed/Asm subroutines. The file on this disk is provided in a LISA 2.5 compatible format (which can also be read in by LISA version 3.0). For those who are using an assembler other than the LISA interactive assembler, the SA.EQUATES file is reproduced in the program listing.

Incidentally, if you're a beginner and in the process of choosing an assembler, I would highly recommend a look at LISA. It is an interactive assembler that makes learning assembly language much easier. Unlike other assemblers, LISA catches errors on input, much like Basic.

I'm going to make the assumption, in this column, that you're using the LISA assembler. Attempting to describe every LISA feature to users of other assemblers would be too great a distraction. Attempting to write the code in a general fashion would make the programs overly large and confusing.

The remaining six programs provided on the Speed/Asm disk are test

Address correspondence to Randall Hyde, Lazer Microsystems, 1791 Capital, Corona, CA 91720.

programs that were used to help validate the Speed/Asm package. You may want to look at these files to get an idea of how Speed/Asm programs are written.

Preparing for Your First Program

Before we can jump in and run our first program there are several decisions that must be made. To begin with, we must decide where the program will run. Unlike Basic, which uses the same data in memory for the source code and run-time code, assembly language programs must be converted from a *source* (human-readable) format to an *object* code (computer-readable) format. This conversion is accomplished by an assembler like LISA. Once the source file is converted to object code you can run the program by executing the object code.

Most assemblers on the market (including LISA) operate in a *co-resident* mode. This means that the source text file and object code both reside in memory during the assembly of the program. Since both files are maintained in RAM at the same time, care must be taken to ensure that the source and run-time object code do not disturb one another. In Basic, memory was allocated automatically for you. While using 6502 assembly language, the memory management chore is left up to the programmer. So extra care must be taken when creating programs.

By referring to Figure 1 you can get an idea of what the Apple's memory space looks like while using LISA. In particular, locations $800 through $17FF are reserved for holding your object code, and locations $1800 and up are reserved for the source file. Four kilobytes of RAM should prove to be sufficient for most programs. Advanced programmers who require more RAM should consult the LISA documentation to find out how to adjust the default textfile/object code size settings.

Whenever you assemble a program, LISA automatically begins assembling and storing the code at location $800. Unless you want to assemble the code at some other location, no special action is necessary to assemble your code into the proper space in RAM.

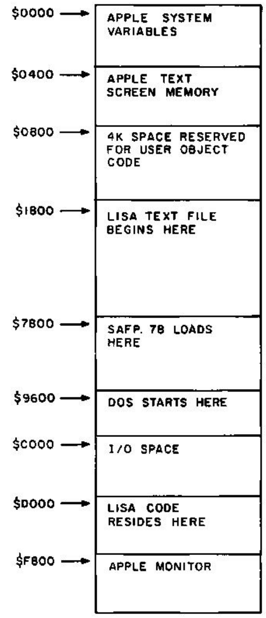


Figure 1. Apple memory map.

Writing Your First Assembly Language Program

The first program we'll write will do nothing at all! Actually it does do something: it immediately returns control to the Apple monitor. This program, as simple as it is, is important because it demonstrates how to terminate a 6502 assembly language program.

Consult the documentation that came with your assembler and learn how to use the editor to enter text into the source file buffer. Once you've mastered the editor, enter the following program into the text buffer:

```
EXIT  EQU  $FF59
      JMP  EXIT
      END
```

Once you've entered this program, assemble it using the assembler (the documentation should describe how to assemble a file). With LISA this is accomplished by typing ASM followed by a carriage return. Once the assembler has completed the task of assembling the code, you should access the Apple monitor program (LISA users type BREAK) and issue the Apple moni-

Listing. Speed/Asm Equates.

```
0800            TTL "Listing One: SPEED/ASM Equates"
0800   ;
0800   ; GENERAL PURPOSE EQUATES
0800   ;
0000   FORASAV  EPZ 0
0001   FORXSAV  EPZ FORASAV+1
0002   FORYSAV  EPZ FORXSAV+1
0003   FORZPG   EPZ FORYSAV+1
0005   DESTADR  EPZ FORZPG+2
0007   PTRADR   EPZ DESTADR+2
0009   ISIMMED  EPZ PTRADR+2
000A   OP       EPZ ISIMMED+1
000C   MAXLEN   EPZ OP+2
000D   VALUE    EPZ MAXLEN+1
000F   DIGIT    EPZ VALUE+2
0010   LEAD0    EPZ DIGIT+1
0011   JMPADR   EPZ LEAD0+1
0013   COUNT    EPZ JMPADR+2
0014   GOTLN    EPZ COUNT+1
0015   LINEINDX EPZ GOTLN+1
0016   SIGN     EPZ LINEINDX+1
0017   ACL      EPZ SIGN+1
0018   ACH      EPZ ACL+1
0019   XTNDL    EPZ ACH+1
001A   XTNDH    EPZ XTNDL+1
001B   AUXL     EPZ XTNDH+1
001C   AUXH     EPZ AUXL+1
0800   ;
0033   PROMPT   EPZ $33
004E   RNDL     EPZ $4E
004F   RNDH     EPZ $4F
0100   STACK    EQU $100
0200   INPUT    EQU $200
0800   ;
```

Listing continued.

Listing continued.

```
0000    FALSE     EQU 0
0001    TRUE      EQU 1
008D    CR        EQU $8D
0800    ;
0800    ;
0800    ; "IF" STATEMENT EQUATES
0800    ;
00BD    EQ        EQU "="
00A3    NE        EQU "#"
00BE    GT        EQU ">"
00BC    LT        EQU "<"
BDBE    GE        EQU ">"|"="*256
BDBC    LE        EQU "<"|"="*256
0800    ;
0800    ;
0800    ;
0800    ;
0800    ; SPEED/ASM ENTRY POINTS
0800    ;
0800    ;
0800    ;
0800    ; NOTE: THE EQUATE OF PUTC MUST
0800    ; BE CHANGED IF YOU RELOCATE
0800    ; SPEED/ASM TO SOME LOCATION
0800    ; OTHER THAN $7800
0800    ;
7800    PUTC      EQU $7800
7803    GETC      EQU PUTC+3
7806    SAGL      EQU GETC+3       ;FOR USE BY S/A ONLY- SEE DOC.
7809    SAPC      EQU SAGL+3       ; "   "   "  "   "      "   "
780C    HOME      EQU SAPC+3       ;HOME AND CLEAR
780F    READLN    EQU HOME+3
7812    INIT      EQU READLN+3
7815    FOR       EQU INIT+3
7818    FOR0      EQU FOR+3
781B    NEXT      EQU FOR0+3
781E    IFI       EQU NEXT+3
7821    IFI0      EQU IFI+3
7824    IFS       EQU IFI0+3
7827    IFS0      EQU IFS+3
782A    MOVE      EQU IFS0+3
782D    LOAD      EQU MOVE+3
7830    MOVS      EQU LOAD+3
7833    LDSTR     EQU MOVS+3
7836    PRINT     EQU LDSTR+3
7839    PRTSTR    EQU PRINT+3
783C    PRTINT    EQU PRTSTR+3
783F    RDSTR     EQU PRTINT+3
7842    RDINT     EQU RDSTR+3
7845    ONXGOTO   EQU RDINT+3
7848    CASE      EQU ONXGOTO+3
784B    CASEI     EQU CASE+3
784E    INSET     EQU CASEI+3
7851    NOTINSET  EQU INSET+3
7854    ABS       EQU NOTINSET+3
7857    NEG       EQU ABS+3
785A    MUL       EQU NEG+3
785D    DIV       EQU MUL+3
7860    MOD       EQU DIV+3
7863    RND       EQU MOD+3
7866    SUBSTR    EQU RND+3
7869    INDEX     EQU SUBSTR+3
786C    LENGTH    EQU INDEX+3
786F    CONCAT    EQU LENGTH+3
7872    GETWZPG   EQU CONCAT+3     ;USED BY SPEED/ASM
7875    RDFP      EQU GETWZPG+3
7878    PRTE      EQU RDFP+3
787B    PRTF      EQU PRTE+3
787E    FADD      EQU PRTF+3
7881    FSUB      EQU FADD+3
7884    FMUL      EQU FSUB+3
7887    FDIV      EQU FMUL+3
788A    FLT       EQU FDIV+3
788D    FIX       EQU FLT+3
7890    FNEG      EQU FIX+3
7893    FADDTN    EQU FNEG+3
7896    FSUBTN    EQU FADDTN+3
7899    FTIMES    EQU FSUBTN+3
789C    FINTO     EQU FTIMES+3
789F    IFF       EQU FINTO+3
78A2    MOVFP     EQU IFF+3
0800    ;
0800              END
```

tor command 800G. The computer will beep and return you to the monitor mode. Congratulations! You've just run your first 6502 assembly language program.

A discussion of exactly what happened may help clear up any problems you have understanding the operation of this program. First of all, the statement

```
EXIT EQU $FF59
```

isn't a true 6502 assembly language statement at all. The EQU instruction is an example of a *pseudo opcode*, or *assembler directive*, provided by most assemblers. This instruction tells the assembler to replace every occurrence of EXIT with the value $FF59 during the assembly of the program. So, the next instruction (JMP EXIT) is converted to JMP $FF59.

The JMP (jump) instruction on the second line performs the same operation as a Goto in Basic. The JMP EXIT instruction directs the 6502 to jump to address $FF59 and begin executing code there. The entry point for the Apple monitor program just happens to be at address $FF59, so jumping to this location reactivates the Apple monitor. Since the program is executed from the Apple monitor and immediately returns control to the monitor, it will appear that nothing has happened.

The END pseudo opcode is required at the end of all LISA source files. It marks the physical end of the program. The END statement does *not* terminate the execution of your program. It is simply a marker for the assembler so it knows when to stop assembling the file. To terminate the program jump to location $FF59.

Other Methods of Terminating Your Programs

In addition to jumping to location $FF59, there are three other ways to terminate an assembly language program. Providing you are not within some nested subroutine and you haven't pushed any data onto the stack (advanced stuff), you can return control to the Apple's monitor

Congratulations! You've just run your first 6502 assembly language program.

with the simple command

```
RTS
```

This is a return-from-subroutine instruction. It's quite similar to Return in Basic. This program can be rewritten as:

```
RTS
END
```

If you assemble and run this program, control will once again be returned to the Apple monitor. This time, however, the speaker won't beep at you.

Another method for program termination is the BRK instruction. This instruction (used in a manner identical to RTS) beeps the speaker, prints the contents of the 6502's registers, and then transfers control to the Apple monitor. BRK is used mainly for debugging purposes, but you can use it anytime it's convenient to get a printout of the 6502 registers.

The last method for program termination I'll mention is very similar to the first example. A jump to location $FF69 also transfers control to the Apple monitor. Basic users are already familiar with this entry point to the Apple monitor. It's the CALL-151 instruction you use to get into the monitor in the first place.

Writing Your First Speed/Asm Program

Now that you can stop your program, the next step in learning how to use Speed/Asm is to begin writing programs that use it. Load the SA.EQUATES textfile into LISA and insert the following code just before the END statement in line 121:

```
EXIT  EQU  $FF69
;
;
         JSR  INIT
         JMP  EXIT
```

Assemble the program as before. But before you run it (and while you're still in LISA) type control-D BLOAD SAFP.78. Now break to the monitor and run the program by typing 800G. Note that the program will ask if your Apple can display upper- and lowercase. Once you type Y or N, control returns to the Apple monitor. If you answer Y, then all text sent to the screen is sent completely unmodified. If you answer N, then all lowercase characters are converted to uppercase before being output to the screen. This allows you to take advantage of lowercase add-on equipment, like the Lazer Microsystems' Lower Case +Plus and Keyboard +Plus units, without having to worry about compatibility problems with Apples not so equipped.

The JSR INIT instruction is new. JSR (which stands for jump to subroutine) is quite similar to GOSUB in Basic. Control is transferred to the specified address, where some routine is executed. The program then resumes execution at the next instruction after the JSR, whenever an RTS instruction is executed. The INIT subroutine entry address is defined in the Speed/Asm equates. *Init must be called before running any Speed/Asm program.* Failure to do so will cause the package to malfunction.

Once INIT is through doing its thing (asking you about lowercase and initializing the system), control is returned to the Apple monitor by the execution of the JMP EXIT instruction.

At this point we've mastered all the mechanics of running a Speed/Asm program: begin by JSRing to INIT, and end with a jump to location $FF69. Your Speed/Asm pro-

gram fits in between these two instructions. Consider the following program:

```
EXIT  EQU  $FF69
;
;
      JSR  INIT
      JSR  PRINT
      BYT  "This is a test",CR,0
      JMP  EXIT
```

It prints

```
This is a test
```

followed by a carriage return onto the video screen. This is but a short example of how to write your own Speed/Asm program. I will describe what this program does when I talk about the Print subroutine.

Declaring Variables in Speed/Asm: A Short Course in Data Structures

Speed/Asm programs can use any of four different data types: individual character, character string, integer number, and real (floating point) number. Unlike Basic, but similar to Pascal, storage space for all variables used in a Speed/Asm program must be allocated somewhere in the program. In order to define a variable for use by Speed/Asm you must understand the underlying representations for the character, string, integer, and real data types.

The Apple's memory space consists of 65,535 memory cells called bytes. Each byte holds one character, or other sub-unit of data. String, integer, and floating point variables are created by combining several bytes. For example, integer variables reside in two contiguous bytes of RAM; real variables (in Speed/Asm) require eight contiguous bytes. String variables need n+2 bytes where n is the maximum number of characters you want the string to hold. You must explicitly reserve sufficient space for each variable you plan to use.

Most assemblers provide several pseudo opcodes to be used to reserve blocks of memory for multi-byte data types. Applicable pseudo opcodes provided in the LISA assembler include BYT, HBY, HEX, ADR, DBY, STR, ASC, INV, BLK, .DA, DCI and DFS. Most of them differ in how they allow you to *initialize* the variables you are defining. For our purposes the BYT, ADR, STR and DFS pseudo opcodes will be used to declare variables.

BYT reserves one byte of storage for each operand present on the line after the BYT instruction. For example, the instruction

```
BYT 0,1,2,3,4
```

reserves five bytes of memory and *initializes* them to the values zero, one, two, three and four. Since a single byte can only represent numeric values in the range 0 to 255, any attempt to define an initial value outside this range will be futile.

If you attempt to define a value greater than 255 using the BYT pseudo opcode, LISA uses the low order (L.O.) byte of the value as the initial value for that location. For example, the statement

```
BYT EXIT
```

where EXIT is equated to $FF69 produces the same code as

```
BYT $69
```

The high order (H.O.) of EXIT's value is ignored by the BYT pseudo opcode.

To define a single-byte variable (possibly to hold a character value), place the variable's name in column 1 of the line containing the BYT pseudo opcode. If you want to define a single-byte variable called CHAR, do so using the statement

```
CHAR BYT 0
```

This statement defines CHAR, reserves space for it and initializes the variable to zero.

Another way to reserve space for a variable is with the DFS (define storage) pseudo opcode. Whereas BYT's operand specifies the initial value the space reserved occupies, DFS's operand specifies how many bytes are to be reserved. If you don't need to initialize CHAR to zero, an alternate method of defining it could be

```
CHAR DFS 1
```

This instruction tells LISA to reserve one byte of storage for the CHAR variable.

Integer variables in Speed/Asm are represented using the standard two-byte two's complement format (see Figure 2). Fifteen bits are used to hold the numeric value, and bit number 16 holds the sign. A two-byte integer variable can represent values in the range −32768 to +32767. Basic programmers should quickly recognize this range as the same supported by Basic integer variables. Basic uses this same format to store integer variables.

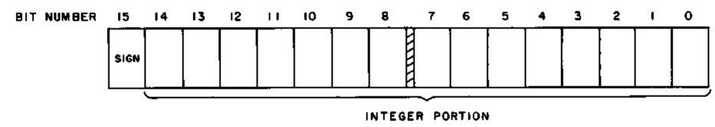


Figure 2. Two's complement representation.

To reserve space for an integer variable in Speed/Asm you must reserve at least two bytes. One way to accomplish this is to follow the BYT pseudo opcode with two operands—that is,

```
INTGR BYT 0,0
```

But this method is inelegant because BYT should be used for declaring byte variables. The ADR pseudo opcode provides a much better alternative to the BYT pseudo opcode for declaring integer variables. The previous declaration could be rewritten as

```
INTGR ADR 0
```

Since *two* bytes are reserved for each operand, only one operand need follow the ADR pseudo opcode. Furthermore, if you follow the ADR pseudo opcode with a value that requires two bytes to hold (like EXIT, or some value greater than 255 or less than zero), ADR will properly store the two's complement repre-



Plug 3,000 new applications into your Apple®

THE CP/M Card™ plugs CP/M Plus™ into your Apple.
The CP/M Card gives you the option of running your Apple II with the speed and capability of a professional Z-80 system with CP/M®-compatible software. You plug in the CP/M Card. Then choose CP/M or your standard Apple software at your option.

Plug into a big, new world of software.
The CP/M Card gives you instant access to the world's largest selection of microcomputer software—more than 3,000 CP/M-compatible applications, languages, and programming utilities. So, you, too can use professional business programs such as WordStar,® SuperCalc,™ Condor,™ and other high-performance software from Day One. Yet, you still have access to your present library of Apple software.

Plug into incredible performance.
Together, the ultra-fast CP/M Card and CP/M Plus run applications up to 300% faster than your Apple system! The CP/M Card is the only Apple II performance package that offers the speed and efficiency of CP/M Plus.

A plug about quality.
The CP/M Card was designed and built by Digital Research, the creators of CP/M, and Advanced Logic Systems, the most respected manufacturer of Apple performance products. So you know the CP/M Card is the most perfectly integrated Apple performance package you can buy.

Why just keep plugging along? The CP/M Card provides everything you need—including 64K of on-board memory, CP/M Plus, CBASIC,® GSX™-80 and full documentation—for just $399.

Now available through the CP/M library.

See your local microcomputer dealer today. Or contact Advanced Logic Systems, 1195 East Arques Ave., Sunnyvale, CA 94086 (800) 538-8177. (In California (408) 730-0306.)

als

Advanced Logic Systems

The CP/M Card for your Apple II.

Also available for the Apple IIe.

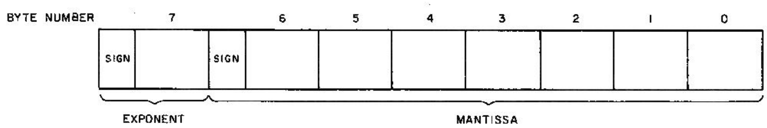
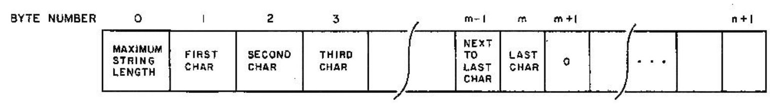


Figure 3. Floating point representation.



Notes:
1) Strings always begin with a byte containing the maximum allowable length of the string.
2) The maximum length is followed by m characters where m is the current dynamic length of the string.
3) All strings are terminated with a zero byte that marks the end of the string.
4) At least n + 2 bytes must be reserved for a string with maximum length n.

Figure 4. String representation.

sentation in memory for you.

Another way to reserve space for an integer variable is with the DFS statement. The instruction

```
INTGR DFS 2
```

reserves two bytes of storage for the integer variable INTGR.

Floating point variables in Speed/Asm require eight bytes. The only practical way to reserve space for a floating point variable is to use the DFS statement. Since eight bytes are required, the definition of a floating point variable takes the form

```
FLTPT DFS 8
```

Since LISA doesn't support a floating point pseudo opcode, you cannot initialize a floating point variable unless you know the hexadecimal representation for the value you're interested in. Figure 3 shows the floating point format used by the Speed/Asm package.

The last data type supported by the Speed/Asm package is the string data type. Strings differ from data of other types in that they vary in length. The amount of space you reserve for a string depends entirely on how big you wish to allow it to grow. A string in Speed/Asm supports two length attributes: a maximum string size and a dynamic (current length) size. Figure 4 shows the format of a string in Speed/Asm.

To declare a string you must reserve n + 2 bytes, where n is the maximum number of characters you wish to allow the string to hold. The first byte of the string must be initialized with the maximum length of the string. For an uninitialized string, the second byte should contain zero. The easiest way to define a string is to use the ADR and DFS pseudo opcodes in conjunction with one another as follows:

```
STRING  ADR  40
        DFS  40
```

The ADR pseudo opcode initializes the first byte of the string to 40 (the maximum length of the string) and the second byte of the string is initialized to zero. The DFS 40 instruction reserves the 40 bytes necessary to hold the string. Additional methods of reserving space for strings will be considered as the need arises.

Placement of Variables In Your Program

One last detail concerning the declaration of variables is the placement of variables within your Speed/Asm program.

The safest place is between the final JMP EXIT instruction and the END pseudo opcode. For technical reasons it is imperative that you do *not* place your variable declarations at the beginning or in the middle of the program.

In this article, the first installment in a series, I've discussed some features of the Speed/Asm package and how to reserve space for variables. Next time I'll describe how to use these variables and write some real Speed/Asm programs. ■

HUNTINGTON COMPUTING



THE HUNTINGTON GANG won't hold you up with high prices. We try to offer you the largest selection in the country at discount prices. Left to right, back row: Danny Huntington, Angela Huntington, Fred Huntington; front row: Barb Huntington (holding Dale Huntington) and Melody Huntington.

SOFTLIGHTS

By Fred Huntington

There are so many new products to tell about that it's hard to know where to begin. Just remember, if its available for Apple or Atari, chances are we either have it in stock or can get it for you.

NEW ATARI STUFF

Play Bingo on the Atari for only **$9.94** cassette (#1580) or disk for **$12.94** (#1581). The new In-Home keyboard (#1215) for the Atari 400 is a class piece of hardware on sale for **$99.44.**

Synapse still is our top-selling company for Atari software. Slime, Claim Jumper, Picnic Paranoia, and Raptilian are just a few of those available for **$29.64** on either cassette or disk.

CLASS ACT: PENGUIN SOFTWARE

We always thought that Penguin Software was among the top available. Their graphics software is absolutely tops. If you read my article in the March **inCider,** you know what I think of high prices of software. Well, my friends at Penguin have done something about those high prices. They have dropped the prices of all their games to **$19.95.**

In addition they've taken some already good games and improved upon them. More for less! To celebrate, we're offering the following specials: #1756 Transyslvania, #1761 Spy's Demise, and #1760 Pie Man all for **$14.99** each! All are excellent games.

A LEGEND IN THEIR OWN TIME

Legend makes some terrific memory expansion boards for the Apple II™. We have special sale prices on their 64K and 128K boards. I have a 128K board in my personal computer and think it's great.

Apple Flasher is a nifty little program that allows you to present an Apple slide show. It's gotten great reviews and only costs **$33.89.**

EDUCATIONAL SOFTWARE

We have more educational software than anybody in the country that we know of. Schools: we will take your purchase orders if prepaid, COD, or if you agree to pay the full list price.

Take a look at the Special Learning Ed Software by Sled for spelling (four disks). List price $230. — on sale for **$179.**

Our best special of the month: Terrapin (#9140) LOGO on sale for **$99.** All PAL software available.

Here's a super program from Avon/Camelot: A Kid's Guide to BASIC Programming BASIC FUN — only **$2.25.** A great book.

Then there's Micro Mother Goose for 3-9 year olds (not copy protected) for **$33.89.**

Finally check out the Apple II™ Reference guide by Nano for **$4.49.** It's a card that has most of the things you'll ever need to know about your Apple — for beginning and advanced programmers.

All the top 30 always available at 15% discount

HUNTINGTON COMPUTING
Post Office Box 1297
Corcoran, California 93212

Foreign Orders 209-992-4481
In California 800-692-4146

Apple™ is a registered trademark of Apple Computer, Inc
Pet™ is a registered trademark of Commodore
TRS-80™ is a registered trademark of Tandy Corp
Atari™ is a registered trademark of Atari, Inc

Outside Calif. 800-344-5106

We take **MasterCard, American Express or VISA** (Include card # and expiration date). California residents add 6% tax. Include $2.00 for postage. Foreign and hardware extra. Foreign (excluding Canada) remit U.S. currency, checks on U.S. banks, use listed charge cards, or make direct wire transfers through Security Pacific Bank, Corcoran, for a $6.00 charge. All overseas orders shipped by air. Send for free catalog. Prices subject to change without notice.

Circle 37 on Reader Service card.

by Gregory R. Glau

Walt's Secret Formula

How would you like to demonstrate visually to your employees how they are performing, and at the same time have for yourself—the boss—some hard figures on their daily production?

Well, your Apple offers a unique way to do just that. And, it's even fun to do!

Surprisingly, survey after survey shows that money isn't the main thing employees consider when asked what's important in their work. It's close to the top, of course, but even more essential is a knowledge of what's happening in the business, whether things are going well or poorly, and how their own contributions affect the company.

What the businessman needs is some method to communicate to his employees...

(1) if the work they are doing is acceptable;

(2) if they are, in fact, making money for the company;

(3) how they can improve, and better yet, *see* their improvement.

And, at the same time, give himself an easy-to-use guide to see problems before they become major difficulties.

This may sound like a large order, but there is a simple solution—one that your Apple can help with.

Now, perhaps you get a Profit and Loss statement every month, or every quarter. This idea is not meant to replace the P & L, but rather to give you an extra tool, one that is specifically designed to use with your employees. It's one thing to say, "Well, folks, we lost $12,000 last quarter" (they may not even believe you), but quite another to have a monthly chart that they can see and easily understand. This tool lets you show your people how they and the business are performing, without divulging actual profit information or amounts.

No fancy program is needed. In fact, you can do the math with a hand calculator. Where your Apple comes into play is in presenting the information to your employees.

You need a program to draw and print a graph, and a word processing program is helpful, too. If you don't have a word processor and/or a graphics system for your Apple yet, head down to your local computer store and write the check. It will be money more than well spent.

--- THIS CHART SHOWS THE DOLLARS PER MAN-HOUR WORKED.
--- BY COMPARING YOUR MONTHLY SALES TOTAL TO THE HOURS YOUR EMPLOYEES WORKED, YOU CAN EASILY ARRIVE AT THE VOLUME PER EMPLOYEE, PER MAN/HOUR.

HOURS WORKED ! ! V   MONTHLY SALES -->

HOURS WORKED	20000	25000	30000	35000	40000	45000	50000	55000	60000
500	40.00	50.00	60.00	70.00	80.00	90.00	100.00	110.00	120.00
550	36.36	45.45	54.55	63.64	72.73	81.82	90.91	100.00	109.09
600	33.33	41.67	50.00	58.33	66.67	75.00	83.33	91.67	100.00
650	30.77	38.46	46.15	53.85	61.54	69.23	76.92	84.62	92.31
700	28.57	35.71	42.86	50.00	57.14	64.29	71.43	78.57	85.71
750	26.67	33.33	40.00	46.67	53.33	60.00	66.67	73.33	80.00
800	25.00	31.25	37.50	43.75	50.00	56.25	62.50	68.75	75.00
850	23.53	29.41	35.29	41.18	47.06	52.94	58.82	64.71	70.59
900	22.22	27.78	33.33	38.89	44.44	50.00	55.56	61.11	66.67
950	21.05	26.32	31.58	36.84	42.11	47.37	52.63	57.89	63.16
1000	20.00	25.00	30.00	35.00	40.00	45.00	50.00	55.00	60.00
1050	19.05	23.81	28.57	33.33	38.10	42.86	47.62	52.38	57.14
1100	18.18	22.73	27.27	31.82	36.36	40.91	45.45	50.00	54.55
1150	17.39	21.74	26.09	30.43	34.78	39.13	43.48	47.83	52.17
1200	16.67	20.83	25.00	29.17	33.33	37.50	41.67	45.83	50.00
1250	16.00	20.00	24.00	28.00	32.00	36.00	40.00	44.00	48.00
1300	15.38	19.23	23.08	26.92	30.77	34.62	38.46	42.31	46.15
1350	14.81	18.52	22.22	25.93	29.63	33.33	37.04	40.74	44.44
1400	14.29	17.86	21.43	25.00	28.57	32.14	35.71	39.29	42.86
1450	13.79	17.24	20.69	24.14	27.59	31.03	34.48	37.93	41.38
1500	13.33	16.67	20.00	23.33	26.67	30.00	33.33	36.67	40.00
1550	12.90	16.13	19.35	22.58	25.81	29.03	32.26	35.48	38.71
1600	12.50	15.63	18.75	21.88	25.00	28.13	31.25	34.38	37.50
1650	12.12	15.15	18.18	21.21	24.24	27.27	30.30	33.33	36.36
1700	11.76	14.71	17.65	20.59	23.53	26.47	29.41	32.35	35.29
1750	11.43	14.29	17.14	20.00	22.86	25.71	28.57	31.43	34.29
1800	11.11	13.89	16.67	19.44	22.22	25.00	27.78	30.56	33.33
1850	10.81	13.51	16.22	18.92	21.62	24.32	27.03	29.73	32.43
1900	10.53	13.16	15.79	18.42	21.05	23.68	26.32	28.95	31.58
1950	10.26	12.82	15.38	17.95	20.51	23.08	25.64	28.21	30.77
2000	10.00	12.50	15.00	17.50	20.00	22.50	25.00	27.50	30.00
2050	9.76	12.20	14.63	17.07	19.51	21.95	24.39	26.83	29.27
2100	9.52	11.90	14.29	16.67	19.05	21.43	23.81	26.19	28.57
2150	9.30	11.63	13.95	16.28	18.60	20.93	23.26	25.58	27.91
2200	9.09	11.36	13.64	15.91	18.18	20.45	22.73	25.00	27.27
2250	8.89	11.11	13.33	15.56	17.78	20.00	22.22	24.44	26.67
2300	8.70	10.87	13.04	15.22	17.39	19.57	21.74	23.91	26.09
2350	8.51	10.64	12.77	14.89	17.02	19.15	21.28	23.40	25.53
2400	8.33	10.42	12.50	14.58	16.67	18.75	20.83	22.92	25.00
2450	8.16	10.20	12.24	14.29	16.33	18.37	20.41	22.45	24.49
2500	8.00	10.00	12.00	14.00	16.00	18.00	20.00	22.00	24.00

--- FOR EXAMPLE, IF YOUR MONTHLY SALES ARE $ 35,000 AND YOUR EMPLOYEES WORKED 1300 HOURS, YOUR DOLLAR PER MAN/HOUR FIGURE IS $ 26.92.

Table 1. Dollars-Per-Person-Per-Hour chart.

Address correspondence to Gregory R. Glau, PO Box 1627, Prescott, AZ 86302.

PUT PRICES IN CHECK

ZIP PACK
RELOAD RIBBONS
FOR
APPLE PRINTERS
NEC 8023A
C. ITOH PROWRITER
$7.99 EA. $89.99 DOZ.

CARTRIDGE RIBBONS FOR
EPSON
MX-80 MX-100
$7.99 EA. $12.95 EA.
$89.99 DOZ. $139.86 DOZ.

MAXELL DISKETTES
5¼ SINGLE SIDE
DUAL DENSITY
MD-1
$29.99 10 PACK

INNOVATIVE CONCEPTS
FLIP'N'FILE
5¼" DISC STORAGE BOX
HOLDS UP TO 60 DISKETTES
$24.95 EA.

RIBBONS FOR
IDS PRINTERS

	EA.	DOZ.
IDS-440	—	$29.99
IDS PAPER TIGER (460/560)	$6.95	$79.23
IDS-PRISM BLACK	$10.95	$124.83

DISKETTE STORAGE BOXES
5¼" — BLUE OR BEIGE
$2.49 EA.

DUAL SPOOL RIBBONS FOR
OKIDATA PRINTERS

	EA.	DOZ.
80, 82, 83	—	$29.99
84	$5.99	$64.69

MEMOREX DISKETTES
5¼ SINGLE SIDE - DUAL DENSITY
#3481
$24.99 10 PACK

COLOR-CODER
LIBRARY CASE SET
CONTAINS 5 BRIGHT COLORS

5¼"	8"
$19.95 SET OF 5	$23.95 SET OF 5

ANTI-STATIC SPRAY
FULL QUART SIZE
WITH DISPENSER
$6.95 QT.
1 GALLON REFILL $19.95

CARTRIDGE RIBBONS FOR
APPLE PRINTERS
NEC 8023A
C. ITOH PROWITER
$9.95 EA. $107.46 DOZ.

LABEL SPECIAL
$2.99 /K
(5K MIN)
1 ACROSS 3½ x 15/16 CONTINUOUS LABELS

MOST RIBBONS AVAILABLE IN COLORS TOO!
CALL OR WRITE FOR OUR SUPPLIES CATALOGUE
ON ORDERS UNDER $14.00 PLEASE ADD $3.00 FOR SHIPPING
MINIMUM RIBBON ORDER $30.00 OR 1 DOZEN

MasterCard VISA

Check-Mate™

TOLL FREE 800-343-7706
IN MASS 617-963-7694
PHONES OPEN 9AM-7PM EASTERN TIME

51 DIAUTO DR. P.O. BOX 103
RANDOLPH, MA 02368

MASS RESIDENTS
ADD 5% SALES TAX

Circle 40 on Reader Service card.

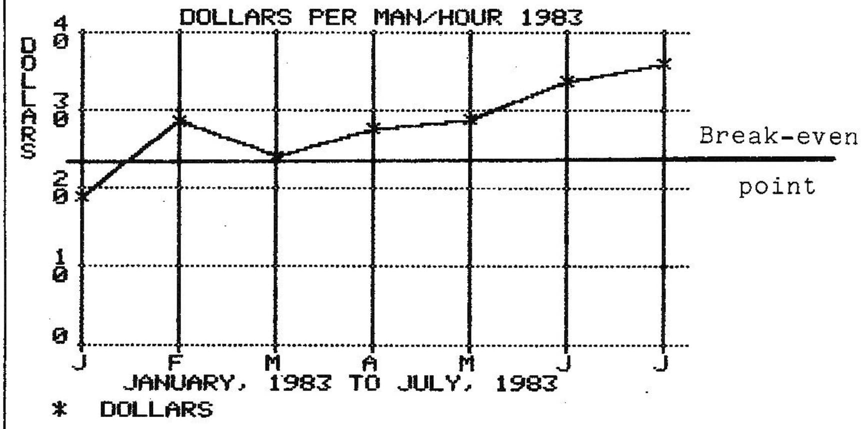


Figure 1. Year-to-date graph.

Dollars-Per-Person-Per-Hour

At the bottom of this is a concept called "dollars-per-person-per-hour" (courtesy of Walt Webb, who has used it for years). It's just what you'd expect—the sales volume produced by each employee, per hour worked.

Many things besides labor affect sales, so you can't look at a single monthly figure and get an honest result. You can graph dollars-per-person-per-hour over a period of time, though, for an excellent guideline. There are many ways this concept can be applied in assessing your firm's financial situation. What is your dollar-per-person-per-hour break-even point? Do you need $20 per hour just to come out even? $30? $40? Any idea? How would you figure it?

If your people worked fewer hours and produced the same volume of sales, what would it do to your profit picture?

If you find your business requires $25 per person per hour just to meet expenses, wouldn't you benefit by checking against that figure every month? Wouldn't that be a good guide as to how your business is doing?

If your dollars-per-person-per-hour starts heading down, doesn't it make sense to *know* about it so you can correct things before you have major difficulties?

Wouldn't it be helpful to see trends of how your employees are doing by graphing monthly dollars-per-person-per-hour?

The place to begin is over the last couple of years. Dig out your last two years' sales data, along with payroll information. Divide monthly sales figures by total hours worked to arrive at a dollar-per-person-per-hour figure for every month, as well as for each year as a whole. The procedure just divides total sales by hours worked.

If your sales were $12,000, and your employees worked 600 hours, your dollars-per-person-per-hour were $20. If you had sales of $50,000, and your people spent 1000 hours on the job, they produced $50 per hour.

Table 1 is a chart that illustrates how the math is done. And, it will help you see the relationship between hours worked and sales volume. Look down the column with the proper amount (you'll have to approximate a bit, obviously) for your monthly sales until it intersects with the total number of hours your people put in for the month. The figure they cross at is your dollar-per-person-per-hour amount. Keep in mind the chart is only a guide, and doesn't cover all sales figures and hours worked. The amount is not important, as every business is different.

The purpose of going back a couple of years to get the monthly figures is not only for comparison (1982 to 1983, for instance), but also to give

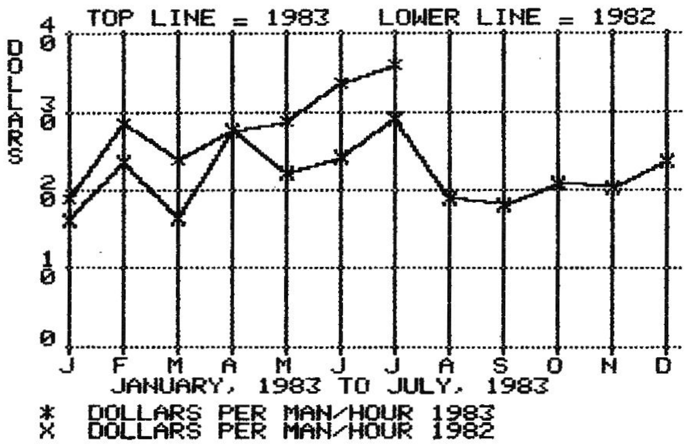


Figure 2. Year comparison graph.

> "Your employees will understand what their contributions mean."

some idea of the dollar amount needed for your business to stay in the black.

Let's say you about broke even one year. That year's dollar-per-person-per-hour figure would give you an idea what you need just to meet expenses. If you had a profit over a year, that dollar-per-person-per-hour figure would suggest the amount you need to make money. If you are running in the red, a comparison of the two would yield an indication of how much you have to improve to be profitable every month.

From the employees' point of view, dollar-per-person-per-hour is easy to understand. They work eight hours, producing X amount of sales. If they can produce more sales in the same time, the business will make more money. And, they will (and should) get a share of it in higher wages and/or profit sharing.

By the same token, if they produce less in sales, you won't make money (may lose it, in fact).

It's important to remember that many things other than wages affect the total sales figure. For instance, if your business sells some "big-ticket" items, they will influence your total sales.

In our air conditioning business, for example, we have a much higher sales volume during the summer months than during the rest of the year. This is because we're selling refrigeration systems, which are expensive items, during this period.

Say one employee spends all of today working on service calls of some sort, but doesn't sell any parts or install any major items. If you charge him out at $30 per hour, his total sales volume is $30 per hour times 8 hours, or $240.

But tomorrow he installs an air conditioning system. He still spends eight hours working, but now you collect $240 for his time, *plus*, say, $800 for the refrigeration equipment, for total sales of $1040.

Quite a difference. When you work with dollars-per-person-per-hour, you have to consider what you sell, and how the time of year affects your sales. You can compare the same months in different years, though.

Also, if you check this dollar-per-person-per-hour figure with other labor versus sales formulas, it will be remarkably consistent over time.

Communicating with Employees

First, though, you have to explain this concept to your employees, so they know exactly what it is. You can present this information in two ways, and it is ideal if you do both.

The first is simply a list of your monthly dollars-per-person-per-hour figures. But it's not enough to post the list on the bulletin board. Write each employee a short note (using your Apple and word processor, of course). Explain the current situation and include a list of monthly dollar-per-person-per-hour amounts for the last year or so.

Mention what the break-even point is, in dollars-per-person-per-hour terms. It will be apparent that if you have a few months of figures lower than what your business needs to stay afloat, there must be changes made in people or methods or both.

The other way to present these figures is in a graph like Figure 1. This is a 1983 year-to-date chart of how the business is doing. Trend lines (is dollars-per-person-per-hour increasing or decreasing?) are immediately apparent. You can draw a line across the graph at your break-even point for a graphic comparison.

A graph, such as Figure 2, that compares years shows employees how they are doing compared to 12 months ago.

Consider the implications for your business if this method works (as it will). Your employees will understand what their contributions mean, they will work harder and you can raise your dollar-per-person-per-hour figure. . . even a dollar or two. If they are working 2000 hours per month, a one dollar rise will provide an extra $2000 per month—$24,000 per year. A two dollar per hour increase will add $48,000 to your yearly sales. If your dollar-per-person-per-hour break-even point is $27, and you can raise it to $29, then $48,000 goes directly into the profit column!

Put this program into action. The two things are all you need to do: write a brief letter to each employee with some past figures, and then every month have your Apple provide the graphs for an update. You'll be pleased with the results. ■

FRIENDLY SERVICE

Business & Utilities

Accent Software
- Soft-Step .... $ 34
- Accu-Shapes .... $ 34
- Loop-Hole .... $ 17
- The Graphic Solution .... $ 95

Applied Software
- Versaform $239

Artsci
- Magic Window II .... $ 95

Ashton-Tate
- dBase II (Req. Z-80) $419

Avant-Garde
- Hi-Res Secrets .... $ 79
- Ultra Plot/Data Graph .... $ 62
- Graphics Application System .... $ 49

Beagle Brothers
- Apple Mechanic .... $ 20
- Tip Disk #1 .... $ 15
- Flex Text .... $ 20
- Frame Up .... $ 20
- Typefaces .... $ 15

Broderbund
- Payroll $249
- Accounts Receivable $249
- The Bank Street Writer .... $ 39

Continental Software
- The Home Accountant .... $ 48
- 1st Class Mail .... $ 48
- CPA Module #1 — G/L $159
- CPA Module #2 — A/R $159
- CPA Module #3 — A/P $159
- CPA Module #4 — Payroll $159
- CPA Module #5 — Property Mgt $159
- The Form Letter .... $ 20

Crane Software
- Menu Generator .... $ 27

Cybertronics
- Karel the Robot $159

Dakin 5
- The Depreciation Planner $249
- The Budget Planner .... $ 95
- The Business Bookkeeping System $249

Datamost
- R.E.I.P. .... $ 82
- Write-On .... $ 82
- Personal Ledger .... $ 82
- The Market Technician .... $ 82
- The Property Manager $189

Datasoft
- Micropainter .... $ 23
- Datasm 65 2.0 .... $ 59
- Lisp Interpreter .... $ 79
- The Basic Compiler .... $ 65

Data Transforms
- Graphtrix 1.3 .... $ 49

Delta Software
- The Bookkeeper Master Program .... $ 59
- The Bookkeeper Check Writer .... $ 27

Denver Software
- Pascal Tutor .... $ 82
- Pascal Programmer $189
- Financial Partner $159

DJR Associates
- FMS-80 (Req. Z-80) $545
- FMS-81 (Req. Z-80) $235
- FMS-82 (Req. Z-80) $235

Don't Ask Software
- S.A.M. .... $ 79

Eagle Software
- Money Decisions $129

Fox & Geller
- Quick Code (Req. Z-80) $189
- Dutil (Req. Z-80) .... $ 62

Hayden Software
- Pie Writer/Multi 80-col. .... $ 95
- Pie Writer/Standard .... $ 95

Hayes Microcomputer Products
- Hayes Terminal Program .... $ 65

Highlands Computer
- E-Z Ledger .... $ 39

Howard Software
- Creative Financing $129
- Real Estate Analyzer II $119
- Tax Preparer 1983 .... $145

Insoft
- Transforth II .... $ 79
- Ald System II .... $ 49
- Graforth II .... $ 49

ISM
- Mathemagic .... $ 59
- Graphmagic .... $ 59

IUS
- Datadex .... $ 99
- Professional Easywriter $115
- Tellstar Level I .... $ 27
- Tellstar Level II .... $ 52
- Forth Development System .... $ 89
- Professional Easymailer .... $ 99
- Original Easywriter .... $ 65
- Original Easymailer .... $ 45
- Easymover .... $ 62
- Pro. Easywriter/Mailer Combo $189
- Orig. Easywriter/Mailer Combo .... $ 89

Kensington Microware
- Format II $159

Link Systems
- Datafax $119
- Datalink .... $ 62
- Link Index .... $ 90
- Link Video .... $ 34
- Link Disk .... $ 45

LJK Enterprises
- Letter Perfect with Mail Merge .... $ 95
- Data Perfect .... $ 69
- Edit 6502 .... $ 69

Lotus
- Executive Briefing System $125

Micro Lab
- The Invoice Factory $129
- The Asset Manager $129
- The Tax Manager .... $ 95
- Data Factory 5.0 .... $189
- Wall Streeter $189
- Payroll Manager $189

Microsoft
- Tasc Compiler $119
- Fortran-80 (Req. Z-80) $125
- A.L.D.S. (Req. Z-80) .... $ 79
- Basic Compiler (Req. Z-80) $249
- Cobol-80 (Req. Z-80) $459
- muLisp/muStar-80 (Req. Z-80) $125
- M-Sort-80 (Req. Z-80) $125
- muMath/muSimp-80 (Req. Z-80) $159
- Time Manager .... $ 95
- Multiplan (Apple Dos or Z-80) $175

Micro-Sparc
- Ampersoft .... $ 34
- Directory Master .... $ 20

COMPUTER OUTLET

Education
- Bumble Plot .... $ 39
- Gertrudes Secrets .... $ 49
- Gertrudes Puzzles .... $ 49
- Rocky's Boots .... $ 49
- Snooper Troops #1 .... $ 30
- Snooper Troops #2 .... $ 30
- Story Maker .... $ 26
- Face Maker .... $ 26
- Compu-Read .... $ 23
- Spelling Bee w/Reading Primer .... $ 27
- Algebra I .... $ 34
- Fractions .... $ 34
- Decimals .... $ 34
- Master Type .... $ 27
- Type Attack .... $ 27
- Wordrace .... $ 17
- Dueling Digits .... $ 20
- SAT Word Attack .... $ 34
- New Step by Step .... $ 59
- Delta Drawing .... $ 45
- Harcourt Brace S.A.T. Series .... $ 59
- Spanish For Traveler .... $ 39
- Whole Brain Spelling .... $ 23
- Speed Reading Courseware .... $ 65
- Speed Read Plus .... $ 39
- PSAT Word Attack Skills .... $ 34
- Perception .... $ 17
- Word Scramble/Super Spell .... $ 15
- U.S. Constitution Tutor .... $ 20
- Algebra II .... $ 20
- Algebra III .... $ 27
- The Linguist .... $ 27
- Know Your Apple .... $ 23
- E-Z Learner .... $ 24
- Crossword Magic .... $ 34
- Counting Bee .... $ 20
- Compu-Math/Arith. Skills .... $ 34
- Planetary Guide .... $ 23
- Star Gazers Guide .... $ 22
- Astro Quotes .... $ 17
- Juggles Rainbow .... $ 30
- Bumble Games .... $ 39

Computer Outlet

Park Place — Upper Level
1095 E. Twain — (702) 796-0296
Las Vegas, Nevada 89109

Call Toll Free **800-634-6766** Order Line Only

Information Order Inquiries (702) 369-5523

We accept Major Credit Cards

Mon.-Fri. 8 A.M.-6 P.M.
Sat. 9 A.M.-5 P.M.
Dealer Inquiries Invited

Business & Utilities

Systems Plus
- Accounting Plus — G/L $239
- Accounting Plus (GL, AP, AR) $499
- Accounting Plus (GL, AP, AR, INV) $629

Muse
- Super Text 40/80 .... $115
- Super Text 40/56/70 (Hi-Res) .... $ 79

Novation
- Deaf Disk .... $ 23
- Software Diskette .... $ 23

On-Line
- Screenwriter II .... $ 82
- The Dictionary .... $ 65
- Lisa 2.5 .... $ 55
- Lisa Educational System .... $ 75
- The Artist .... $ 55
- Screenwriter Professional $129
- The General Manager II $149

PBL Corporation
- Personal Investor .... $ 95

Peachtree
(All Peachtree products require a Z-80 card)
- Series 40 Acc't Packages .... ea. $239
- Series 80 Acc't Packages .... ea. $239
- Peachtext $289
- Spelling Proofreader $179
- Mailing List Manager $209

Penguin Software
- Complete Graphics System .... $ 49
- Special Effects .... $ 27
- Graphics Magician .... $ 39
- Additional Fonts & Char. Sets .... $ 15
- Complete Graphics System II (Apple Tablet Version) .... $ 79
- Special Effects (Apple Tablet Version) .... $ 49

Select
- Selectword Processing Sys. $299

Sensible Software
- Sensible Speller .... $ 79
- Image Printer .... $ 27
- Super Disk Copy III .... $ 20
- Multi Disk Catalog III .... $ 17
- Disk Recovery .... $ 20
- Quickloader .... $ 17
- The Bug .... $ 34

Silicon Valley
- Word Handler II $129
- List Handler .... $ 59
- Dictionary .... $ 79

Stoneware Products
- D.B. Master $145
- D.B. Master Utility Pak #1 .... $ 65
- D.B. Master Utility Pak #2 .... $ 65
- Graphics Processing Sys. (Standard) .... $ 49
- Graphics Processing Sys. (Prof.) $119

Synergistic
- The Data Reporter $159
- Integer Basic Compiler .... $ 95
- Global Program Line Editor .... $ 45
- The Inventory Manager .... $ 95

Software Publishing
- PFS: Report .... $ 59
- PFS .... $ 79
- PFS: Graph .... $ 79

Sorcim/ISA
- Supercalc (Req. Z-80) $179
- Spellguard (Req. Z-80) $125

Southeastern Software
- Data Capture 4.0 .... $ 47
- Data Capture 4.0/80 .... $ 59

Southwestern Data (New)
- Apple Doc .... $ 27
- Ace .... $ 27
- List Master .... $ 27
- Speed Star .... $ 65
- Correspondent .... $ 39
- Z Term (Req. Z-80) .... $ 65
- Z Term "The Professional" (Req. Z-80) .... $ 99
- P Term "The Professional" .... $ 85
- Ascii Express "The Professional" .... $ 85
- Disk Library .... $ 34
- Double Time Printer .... $ 65
- Financial Management System II .... $ 45
- Munch-a-Bug .... $ 34
- Printerographer .... $ 34
- Routine Machine .... $ 45

Visicorp
- Visicalc 3.3 .... $165
- Visiplot $139
- Visitrend/Plot $199
- Visidex $165

Circle 254 on Reader Service card.

TOP SELLERS

Entertainment

Snapper	$20
Toddlers Tutor	$16
Cyborg	$22
Congo	$23
Goldrush	$23
Alien Lander	$20
Retro-Ball	$20
Space Adventure	$20
International Grand Prix	$20
Conglomerates Collide	$27
Battlesight	$27
Starship Commander	$20
Amoeba Man	$23
The Gauntlet	$23
Crisis Mountain	$23
Nightmare Gallery	$20
U-Boat Command	$20
Escape From Arcturus	$20
Atlantis	$27
Odyssey	$20
Epidemic	$23
Galactic Adventure	$39
Cytron Masters	$27
Galactic Gladiator	$27
The Cosmic Balance	$27
The Battle of Shiloh	$27
S.E.U.I.S.	$27
Guadacanal	$39
Pursuit of the Graf Spree	$39
The Shattered Alliance	$39
Operation Apocalypse	$39
Computer Bismarck	$39
Napoleon's Campaign	$39
"The Alien Game"	$23
Norad	$23
Knight of Diamonds	$23
The Blade of Black Pool	$27
Way Out	$27
Space Eggs	$20
Phantom's Five	$20
Pulsar II	$20
Sneakers	$20
Outpost	$20

SYSCOM II 64K . . $799
(Apple Compatible)
or
OUR SPECIAL PACKAGE
Syscom II 64K, NEC Green Screen Monitor, Fourth Dimension Drive with Controller, Monitor Stand, System Saver Fan, Word Handler Word Processing Package **$1599**

Mystery House	$17
Mission: Asteroid	$15
Frazzle	$17
Castles of Darkness	$23
Acey Deucey	$20
Zargs	$23
Spider Raid	$20
Zork III	$27
Juggler	$20
Mummy's Curse	$20
Kamikaze	$23
Shuttle Intercepts	$23
Horizon V	$23
Lazer Silk	$20
Zenith	$23
High Orbit	$20
Phaser Fire	$20
Prisoner II	$23
Rendezvous	$27
Empire II Interstellar Sharks	$23
Rubicks Cube Unlocked	$17
Wordrace	$17
Zaxxon	$27
Fathom's Forty	$23
Solitaire & Cribbage	$23
Mars Cars	$20
Money Muncher	$20
Crazy Mazey	$20
Raster Blaster	$20
S.A.G.A. Adventures	ea. $27
Serpentine	$23
Choplifter	$23
Frogger	$23
Sea Fox	$20
Hard Hat Noah	$20

EVERYDAY LOW PRICES

PERIPHERALS

Softcard	$ 229
Ramcard	$ 69
Microsoft Premium Sys (Contains Softcard, Ramcard, Videx Videoterm)	$ 479
Fortran 80	$ 139
System Saver Fan	$ 65
Flip & File Diskette Box	$ 21
Cobol-80	$ 449
Extended muMath	$ 169
Enhancer II	$ 99
EZ Port	$ 19
Game Paddles	$ 29
Joystick	$ 39
Select-a-Port	$ 39
Lower Case Adapter	$ 25
Mannesman Tally 160	$ 799
Citoh Prowriter	$ 439
Microtek Apple Parallel Interface	$ 79
Apple Adapter (Wico Joysticks)	$ 17
Wico Joystick	$ 23
Wico Redball	$ 24
Wico Trackball	$ 49
12 Foot Ext. Cord	$ 9
Microbuffer II 16K Parallel	$ 169
Microbuffer II 32K Parallel	$ 199
Kraft Joystick	$ 45
Kraft Paddles	$ 34
Apple Dumpling GX	$ 95
Buffered Dumpling 16K	$ 155
Buffered Dumpling 32K	$ 199
8088 Coprocessor Board	$ 549
The Joyport	$ 34
TG Trak Ball	$ 45
Versawriter Graphics Tablet	$ 209
Videoterm 80 Column Board	$ 229
Vision 80 Board	$ 189
IDS 480 Printer	$ 529
IDS Prism 132 Printer	$1299
Amdek Amber Monitor	$ 179
Amdek Digital Plotter	$ 729
Amdek 3" Micro Floppy Disk Drive	$ 699
Microline 84P	$1029
Microline 83A	$ 679
Microline 82A	$ 429
Daisywriter Letter Quality Printer	$1129
Corona Starfire 5 MB Drive	$1850
Corona Starfire 10 MB Drive	$2289

COMPUTER OUTLET

*** SPECIALS OF THE MONTH ***

**Elephant Disks (Box)	$ 20**
**Amdek Color I Monitor	$ 309**
**Micromodem II	$ 259**
**Novation Apple Cat II Modem	$ 329**
**The Grappler Plus	$ 119**
**NEC 8023A Printer	$ 459**
**Rana Systems 40 Track Disk Drive	$ 299**
**Rana Systems 80 Track Disk Drive	$ 659**
**The Word Handler	$ 129**
**NEC 3530 Printer	$1599**
**Qume Sprint 11 + Printer	$1359**
**Hayes Smartmodem 300	$ 209**
**Amdek Color II Monitor	$ 659**
**Hayes Smartmodem 1200	$ 499**
**Citoh Prowriter	$ 419**
**Screenwriter II	$ 82**

Computer Outlet

Call Toll Free **800-634-6766** Order Line Only

Information Order Inquiries (702) 369-5523

ORDERING INFORMATION AND TERMS:

For Fast Delivery send cashier checks, money orders or direct bank wire transfers. Personal and company checks allow 3 weeks to clear. C.O.D. orders ($3.00 minimum) and 1% of all orders over $300. School purchase orders welcomed. Prices reflect a cash discount only and are subject to change. Please enclose your phone number with any orders. Shipping — Software ($2.50 minimum). Shipping — Hardware (please call). Foreign orders, APO & FPO orders — $10 minimum and 15% of all orders over $100. Nevada residents add 5¾% sales tax. All goods are new and include factory warranty. Due to our low prices, all sales are final. All returns must have a return authorization number. Call 702-369-5523 to obtain one before returning goods for replacement. All returned merchandise is subject to a restocking fee and must come with their original packaging in order to be accepted.

NO returns permitted after 21 days from shipping date.

TOP SELLERS

Entertainment

Fly Wars	$20
Cyclod	$20
Kabul Spy	$20
Zero Gravity Pinball	$20
Copts and Robbers	$23
Escape from Rungistan	$20
Lemmings	$20
Computer Foosball	$20
The Battle of Shiloh	$27
Electric Duet	$20
Temple of Apshai	$27
Tuesday Morning Quarterback	$20
Hi-Res Computer Golf	$20
David's Midnight Magic	$23
Track Attack	$20
Apple Panic	$20
Ultima	$27
Bug Attack	$20
Snack Attack	$20
Thief	$20
County Fair	$20
Compu-Read	$20
Empire I World Builders	$22
Firebird	$20
Sargon II	$23
Zork I	$27
Zork II	$27
Pool 1.5	$23
Castle Wolfenstein	$20
Threshold	$27
Mousekattack	$23
Hi-Res Football	$27
Hi-Res Soccer	$20
Time Zone	$65
Wiz & Princess	$22
Ulysses & The Golden Fleece	$23
Wizardry	$34
Hadron	$23
Beer Run	$20
Gorgon	$27
Photar	$20
Warp Factor	$27
Tigers in the Snow	$27
Computer Baseball	$27
Ceiling Zero	$20
Sherwood Forest	$23
The Queen of Phobos	$23
Bez Wars	$19
Bez Man	$16
Star Blazer	$22
Cross Country Rallye	$20
Swashbuckler	$23
Tumble Bugs	$20
Rear Guard	$20
Hungry Boy	$17
Dneiper River Line	$20
Labyrinth	$20
Oil Rig	$27
Human Fly	$27
Deadline	$34
Minotaur	$23
Bandits	$23
Oil Barons	$65
Monster Maze	$20
New World	$20
GFS Sorceress	$23
Telengard	$19
Starcross	$27
Aztec	$27
Tubeway	$23
Vegas Video	$20
Odin	$34
Star Maze	$23
Zendar	$20
Front Line	$20
Mask of the Sun	$27
Pest Patrol	$20
Lunar Leeper	$20
Pie Man	$20
Thorolian Tunnels	$20
Canyon Climber	$20
Alien	$19
Normandy	$27
Germany	$39
Miner 2049er	$29
AE	$20
Repton	$27
Moon Shuttle	$27
Spy's Demise	$20
Space Vikings	$34
Flip Out	$20
Wavy Navy	$23
Bomb Alley	$39
Ultima II	$39

Circle 254 on Reader Service card.

Solving Problems With Logo

Logo is a powerful language with capabilities much broader than Turtle graphics.

by Greg Stone

"Turtle graphics are fun," said my friend, "but can you write real programs in Logo?"

Of course "real" programs can be written involving turtle graphics, but the question speaks to an honest problem: Logo is too frequently associated only with turtle graphics and kids and too often the full power of the language is ignored.

I wouldn't say that the following program demonstrates the full power of Logo, but it does show how a familiar programming problem, creating a bubble sort, can be executed in Logo. In particular, it demonstrates how Logo handles lists.

The idea of a bubble sort is simple. Compare numbers in a list one at a time in a manner that allows the desired numbers to "bubble up" to the top of the list.

My first attempts to write a Logo program to bubble sort failed because I tried to translate literally between Applesoft Basic and Apple Logo. I simply took a sort routine that used a data statement and arrays and tried to substitute Logo commands for the Basic ones. It quickly became apparent that this was the wrong approach, so I went back to zero and redefined the task. However, I still had a program that

```
TO BUBBLE.SORT
PRINT []
MAKE "A []
INSTRUCTIONS
GET.NUMBERS
SORT
DISPLAY
END


TO INSTRUCTIONS
PRINT []
PRINT [THIS PROGRAM WILL SORT NUMBERS]
PRINT [FROM LOWEST TO HIGHEST.]
PRINT []
PRINT [TO START, ENTER YOUR NUMBERS WITH A]
PRINT [SPACE FOLLOWING EACH NUMBER. THE LAST]
PRINT [NUMBER ENTERED SHOULD BE "99999"]
PRINT []
PRINT [( IF YOU RUN OUT OF SPACE WHEN ENTERING]
PRINT [NUMBERS, SIMPLY HIT RETURN AND]
PRINT [CONTINUE TO ENTER NUMBERS. )]
PRINT [- - - - - - - - - - - - - - - - - - - - -]
PRINT []
END


TO GET.NUMBERS
PRINT []
MAKE "A SENTENCE :A READLIST
IF LAST :A < 99999 [GET.NUMBERS]
END


?PO "SORT
TO SORT
MAKE "C 0
TRY
MAKE "A SENTENCE ( BUTFIRST :A ) ( ITEM 1 :A )
IF :C > 0 [SORT]
END

?PO "TRY
TO TRY
TEST ITEM 1 :A > ITEM 1 BUTFIRST :A
IFTRUE [EXCHANGER]
IFFALSE [MAKE "A SENTENCE ( BUTFIRST :A ) ( ITEM 1:A )]
IF ITEM 1 :A < 99999 [TRY]
END
```

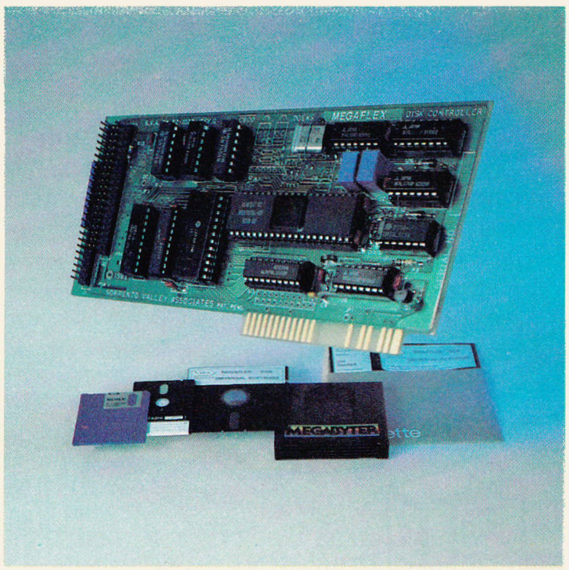
Listing. A bubble sort.

Address correspondence to Greg Stone, 1346 Drift Road, Westport, MA 02790.

Circle 21 on Reader Service card.

MEGAFLEX® ABILITY

You Pick The Disk System, MegaFlex Controls It!



WITH SOFTDRIVERS FOR A FLEXIBLE FUTURE!

MEGAFLEX—a universal floppy disk controller and modern alternative to the Apple drive system offering increased storage, improved reliability and . . . FLEXIBILITY.

Enjoy megabytes of online storage with your choice of micro, mini, or maxi drives—or even 6Mb with the Amlyn cartridge pack! Ideal for high-capacity storage now, winchester-disk backup later.

The MEGAFLEX secret is to autoboot softdrivers that match the needs of your drive system. All hardware functions are software-controlled. MEGAFLEX can match new drive capabilities without hardware changes. Drive-dependent ROMs have been eliminated.

APPLE *///*? OF COURSE!!

MEGAFLEX is compatible with BASIC, CP/M, Pascal, VISICALC, SOS and DOS-emulation on the Apple III, Apple II, Franklin Ace and Basis. All language features and operating system commands (LOAD, BRUN, etc.) are standard. If you can operate Apple drives you can operate MEGAFLEX! Your Apple software will run without modification too.

BRIDGE THE APPLE FORMAT BARRIER!

The MEGAFLEX diskette does what Apple's cannot—read and write diskettes from other computers! Software-controlled industry-standard IBM 3740 or System 34 type formats allow the MEGAFLEX library of reformatting software to read and write Altos, Radio Shack, Osborne, and IBM PC diskettes. (Call for the latest software details.)

MORE STORAGE, MORE UNIVERSAL FEATURES, LOWEST COST

MEGAFLEX with 8″ maxi or high density 5.25″ minis gives you 1.2 Megabyte of formatted data per diskette for 8 times the file and data size!

MEGAFLEX offers flexible software choices:

- data rate (250/500 Kbits per second),
- single and double density recording, and
- single/double sided drive operation (max 4 drives).

MEGAFLEX has the lowest chip count of any controller today! This means less power, a cooler Apple and better reliability.

Lowest price, highest performance, that's MEGAFLEX!

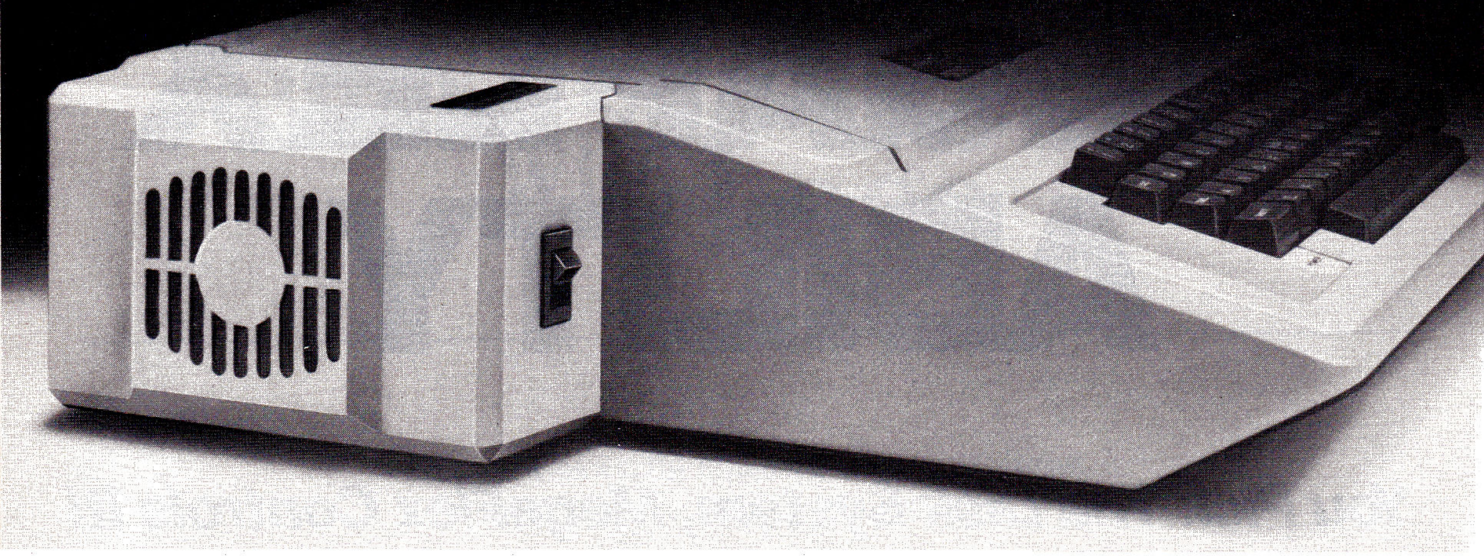
MEGABYTER™
A Division of SVA

11722 SORRENTO VALLEY ROAD
SAN DIEGO, CA
(619) 452-0101
TWX 910-335-2047 APPLE TWO SDG

Circle 86 on Reader Service card.

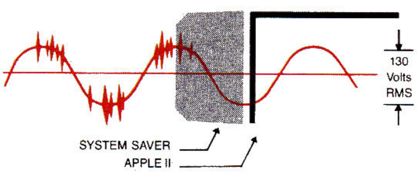
System Saver™

The most important peripheral for your Apple® II.



For Line Surge Suppression

The SYSTEM SAVER provides essential protection to hardware and data from dangerous power surges and spikes.

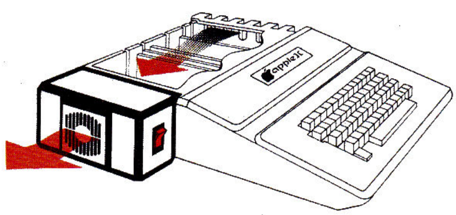


By connecting the Apple II power input through the SYSTEM SAVER, power is controlled in two ways: 1) Dangerous voltage spikes are clipped off at a safe 130 Volts RMS/175 Volts dc level. 2) High frequency noise is smoothed out before reaching the Apple II. A PI type filter attenuates common mode noise signals by a minimum of 30 dB from 600 khz to 20 mhz, with a maximum attenuation of 50 dB.

For Cooling

As soon as you move to 64K RAM or 80 columns on your Apple II you need SYSTEM SAVER.

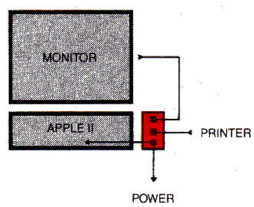
Today's advanced peripheral cards generate more heat. In addition, the cards block any natural air flow through the Apple II creating high temperature conditions that substantially reduce the life of the cards and the computer itself.

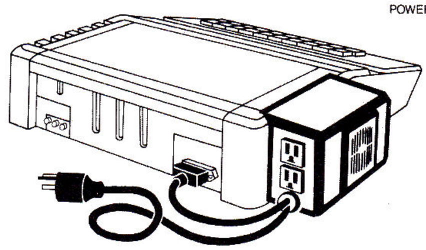


SYSTEM SAVER provides correct cooling. An efficient, quiet fan draws fresh air across the mother board, over the power supply and out the side ventilation slots.

For Operating Efficiency

SYSTEM SAVER contains two switched power outlets. As shown in the diagram, the SYSTEM SAVER efficiently organizes your system so that one convenient, front mounted power switch controls SYSTEM SAVER, Apple II, monitor and printer.

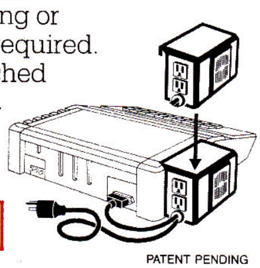




The heavy duty switch has a pilot light to alert when system is on. You'll never use the Apple power switch again!

Easy Installation

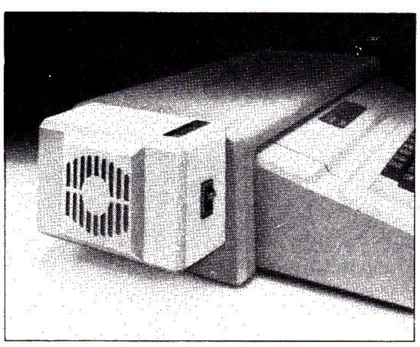
Just clips on.
No mounting or hardware required.
Color matched to Apple II.



UL LISTED

PATENT PENDING

Compatible with Apple Stand



Suggested Retail
One Year Warranty
$89⁹⁵

Kensington Microware Ltd.
300 East 54 Street, Suite 3L
New York, NY 10022
(212) 486-2802

KENSINGTON MICROWARE

Listing continued.

```
?PO "EXCHANGER
TO EXCHANGER
MAKE "A ( SENTENCE ( ITEM 1 :A ) ( BUTFIRST BUTFIRST :A ) ( ITEM 2 :A ) )
MAKE "C :C + 1
END

?PO "DISPLAY
TO DISPLAY
PRINT []
PRINT [THE SORTED LIST IS:]
PRINT []
PRINT BUTLAST :A
END
```

depended on a pair of nested loops, thus showing its Basic heritage, so I tried once more, this time shifting my mind into Logo.

The result can be found in the program listing. The program as written sorts from lowest to highest, although it is easy to reverse. If you have Apple Logo you may want to stop now and type the program into your computer, then read on. If not, just follow along. All Logo commands won't be explained here, but we will dissect the essential ones.

The heart of the program is in the Sort procedure, the procedure it calls, Try, and the procedure it in turn calls, Exchanger. These three do all the work by manipulating a list. The other procedures allow for reminding humans how to use the program (Instructions), inputting a list (Get.numbers) and printing the result (Display). The title procedure, Bubble.sort, merely sets the list at 0 at the start of each use and provides the sequence for calling the other procedures.

Checking It Out

The best way to see what happens is to run a short list of numbers through the program, procedure by procedure. (One of the nice things about Logo programming is it's easy to break things down this way into small, manageable units and let them work independently to see what is happening.)

```
TO BUBBLE.SORT
PRINT []
MAKE "A []
INSTRUCTIONS
GET.NUMBERS
SORT
DISPLAY
END
```

The first line is cosmetic, merely providing a space. The next line creates an empty list labelled A, a necessary step. The program then calls the Instructions procedure. The major point in the instructions is to ask the user to insert a stop code, in this case 99999, at the end of the list of numbers to be sorted.

Once instructions are printed, control returns to Bubble.sort, which then calls the procedure Get.numbers. If you are following on a computer, now's the time to create an empty list by typing MAKE "A []. Then type GET.NUMBERS. You will be greeted by a flashing cursor. Type the following list of numbers to sort:

3 2 1 99999

As mentioned, the 99999 is needed to let the computer know it is at the end of the list.

```
TO GET.NUMBERS
PRINT []
MAKE "A SENTENCE :A READLIST
IF LAST :A<99999 [GET.NUMBERS]
END
```

The second line is the key one. It combines the empty list that we started with, with whatever numbers the user types. *Sentence* is the Logo command for combining elements into a list and *Readlist* is the command for taking inputs from the keyboard.

Why not just say *Make "A Readlist* and forget all this business of an empty list? In truth, that's how the program was written first. The problem lies in not being able to key in more than about 3½ lines before the Apple demands a carriage return. This would limit the size of the list to be sorted.

As written now, when the user hits a carriage return the next line in Get.numbers checks to see if the list is finished. If the last number is less than 99999, it knows the list isn't finished and the procedure calls itself. This is a simple example of recursion. By calling itself this procedure allows you to keep entering numbers into list A indefinitely, or at least until the Apple's memory capacity, or your patience, is exceeded.

If you want to make sure your list is in there, simply type PRINT :A and the result should be 3 2 1 99999. At this point the control shifts back to the next line of Bubble.sort, which calls the procedure Sort.

```
TO SORT
MAKE "C 0
TRY
MAKE "A SENTENCE ( BUTFIRST :A )
  ( ITEM1:A )
IF :C>0 [SORT]
END
```

Sort's first task is to set a counter, C, to 0. That done, it promptly calls the procedure Try, the real workhorse of the program.

```
TO TRY
TEST ITEM 1 :A>ITEM 1 BUTFIRST :A
IFTRUE [EXCHANGER]
IFFALSE [MAKE "A SENTENCE ( BUT
  FIRST :A ) ( ITEM 1:A )]
IF ITEM 1 :A<99999 [TRY]
END
```

I wanted to call this procedure Test, but that word is a Logo primitive. The first line in Try compares the first number in the list with the second number. That is, it tests the truth of the statement that Item 1 is greater than Item 2.

If this statement is true, it then calls the procedure Exchanger. In our example Item 1 (3) is greater than Item 2 (2), so let's move to Exchanger.

```
TO EXCHANGER
MAKE "A ( SENTENCE ( ITEM 1 :A ) ( BUT
  FIRST BUTFIRST :A ) ( ITEM 2 :A ) )
MAKE "C :C + 1
END
```

If you are following on the computer, you have a list that looks like this:

3 2 1 99999

Now type EXCHANGER, and when the prompt reappears, type: PRINT :A, and the computer should respond:

3 1 99999 2

What did Exchanger do? It kicked the 2 to the end of the list so that Try could now test the 3 against the 1. Exchanger does that rather simple task

with the most complicated line of code in the program:

MAKE "A (SENTENCE (ITEM 1 :A) (BUT FIRST BUTFIRST :A) (ITEM 2 :A).

But our list isn't sorted yet. It looks like this:

99999 2 1 3

Since the first number is 99999, control moves from Try to the procedure that called it, Sort.

Sort has three tasks:

1. It kicks 99999 to the end of the list. (It does this first because when Try hands control back to Sort it does so at the line immediately following Try.)

2. It checks the counter, C, to see if any exchanges have been made. If C is greater than 0, then exchanges have been made. This is a signal that the sorting is not completed, so once more we rely on recursion and have Sort call itself. (If :C>0 [Sort].)

3. Before moving to Try with the revised list, Sort resets the counter to 0.

Now Try takes over once more. The list it has looks like this:

2 1 3 99999

A few more passes through the wringer and Try will have a list where each test will prove false and no exchanges will have been made. It will

"Butfirst allows us to work with lists without knowing their lengths."

then hand this list to Sort. In our example it will look like this:

99999 1 2 3

Sort will kick the 99999 to the end of the list. It will then find that C equals 0 (remember, no exchanges were made because the list was in correct order) and instead of calling itself, it will hand control back to the main program, Bubble.sort. This program moves to the next line following Sort, which is Display.

All display does is print a message, "The sorted list is:", as well as the list itself, minus the 99999. This last is taken off by the line: "Print Butlast :A" which, of course, prints everything but the last number in the list.

Picking this apart and using our example (3 2 1 99999), we find the line creates a new list (still called A) which consists of Item 1 in the old list (3), the Butfirst Butfirst of the old list (1 99999) and Item 2 (2) of the old list. (The Butfirst of a Logo list is the entire list but the first number. It follows then that the Butfirst Butfirst is the entire list but the first two numbers. It may sound awkward, but it's really a very handy device that allows us to work with lists without knowing their lengths.)

Incidentally, the parentheses before Sentence and at the end of the line are needed. The other paren-

"Like all bubble sorts, it is slow... Still, for smaller applications it is fine."

theses are there to make the line more understandable to humans, not computers.

After Exchanger has reordered the list, it increments a counter, C, by one. This is a way of keeping track of whether any exchanges have been made. Its usefulness will become apparent later.

Program control now returns to Try, where our list will continue to move until it gets to the stop code, 99999. To demonstrate this, type TRY. When the cursor returns ask to look at the list again by typing: PRINT :A. The result should be:

```
99999 2 1 3
```

This is hardly a sorted list. What has happened is the list has gone through once. When Try tested 3 to see if it was larger than 99999, the answer was "false." That moved control to the line

```
IFFALSE [MAKE "A SENTENCE (BUT
FIRST :A) (ITEM 1 :A)]
```

Since the list was 3 99999 2 1, this line makes a new list, using the Butfirst of the old (99999 2 1) and Item 1 of the old (3).

The next line in Try looks at this new list and, in effect, asks if it is at the end:

```
IF ITEM 1 :A<99999 (TRY)
```

When the first number in the list is 99999, Try stops. If it is anything less than 99999, then Try continues by calling itself.

Pizza?

If you're still with us, type in the entire program and try it by typing BUBBLE.SORT, and following the directions. Don't, however, get too ambitious about the quantity of numbers you want sorted. The program will handle ten numbers in about six seconds. But like all bubble sorts, it is slow. It will take about a minute to sort 25 numbers and, if you want it to handle 100, go out and get a pizza or play a game of chess. It took nearly 20 minutes to handle one list of 110 numbers!

Still, for smaller applications it is fine. If you want to work it into another program, all you need are the procedures, Sort, Try and Exchanger, plus a way of entering numbers and of getting the results out.

To make the program sort from highest to lowest, change your stop code to −99999 at all points. Change the > sign in the TEST line in Try to <, and change the < sign before 99999 (also in Try) to >. Reverse the < sign in Get.numbers. Obviously if you're sorting numbers larger than 99999 or smaller than −99999 you will have to change this stop code to an appropriate value.

The program can stand alone, but next month we'll show how to integrate it with other Logo procedures to perform specific tasks. ■

Dandy DOS Enhancer

DOS enhancers seem to be flooding the market these days. To help you choose which enhancer best suits your needs, stay tuned. This month Diversi-DOS boots up.

by Steven A. Schwartz

Diversi-DOS offers three useful utilities, two of which are typically sold as hardware add-ons at a much higher price. They include: (1) rapid file processing (disk access), (2) a keyboard buffer for type-ahead, and (3) a print buffer. The routines may be included in your programs and distributed without any additional fees to DSR—as long as a special notice is included.

Diversi-DOS is completely menu-driven (see Figure 1). Option 1, "Instructions," provides the necessary information to use the routines. The instructions can be routed to your printer for a permanent copy or, if you prefer, relevant sections can be selected for screen viewing as they are needed. Since the menu options are so simple to use, a more extensive printed manual is unnecessary.

Diversi-DOS Functions

DOS speed-up. DSR claims that disk operations are completed 2–4 times faster than standard DOS 3.3. I ignored these figures, ran a few tests of my own, and got the results shown in the table.

Address correspondence to Dr. Steven A. Schwartz, 9226 Vantine St., Pittsburgh, PA 15235.

Disk access from within programs is also improved—at least with sequential text files. A Read of 500 random data values from a 45-sector text file took 114 seconds with standard DOS, but only 46 seconds with Diversi-DOS. The instruction sheet also shows a "fast" way to read/write random access records. The result is a 60 msec. improvement in speed. I'm afraid I'll have to take DSR's word on this; the difference is so small that I have neither the tools nor the understanding of machine language (e.g., clock cycles per instruction) to substantiate it.

Diversi-DOS skips the Verify procedure that DOS 3.3 normally performs, thus saving access time. If Verify is reestablished, saving the 87-sector Applesoft program above now takes 12 seconds rather than 6. The improvement over DOS 3.3, however, remains substantial.

Adding Diversi-DOS to your program disks is easy; just select menu option 2 and insert your disk. Or, you may select option 7 (EXIT TO BASIC), insert a blank disk, and type INIT HELLO. In either case, the speed feature will be in effect every time you boot the disk.

Menu option 3 allows you to use Diversi-DOS with protected commercial programs. Option 3 tells you the name of the Hello program on the protected disk. You may then write a short routine (1–3 lines) to allow the program to run under Diversi-DOS control. Unfortunately, my test runs with graphic game programs were unsuccessful. If you still wish to add Diversi-DOS to the disk, DSR suggests that you contact the manufacturer—not them.

The only limitation I uncovered was that it may not function properly with programs that also modify DOS. Overall, this feature is impressive and, in some instances, even surpasses DSR's claims.

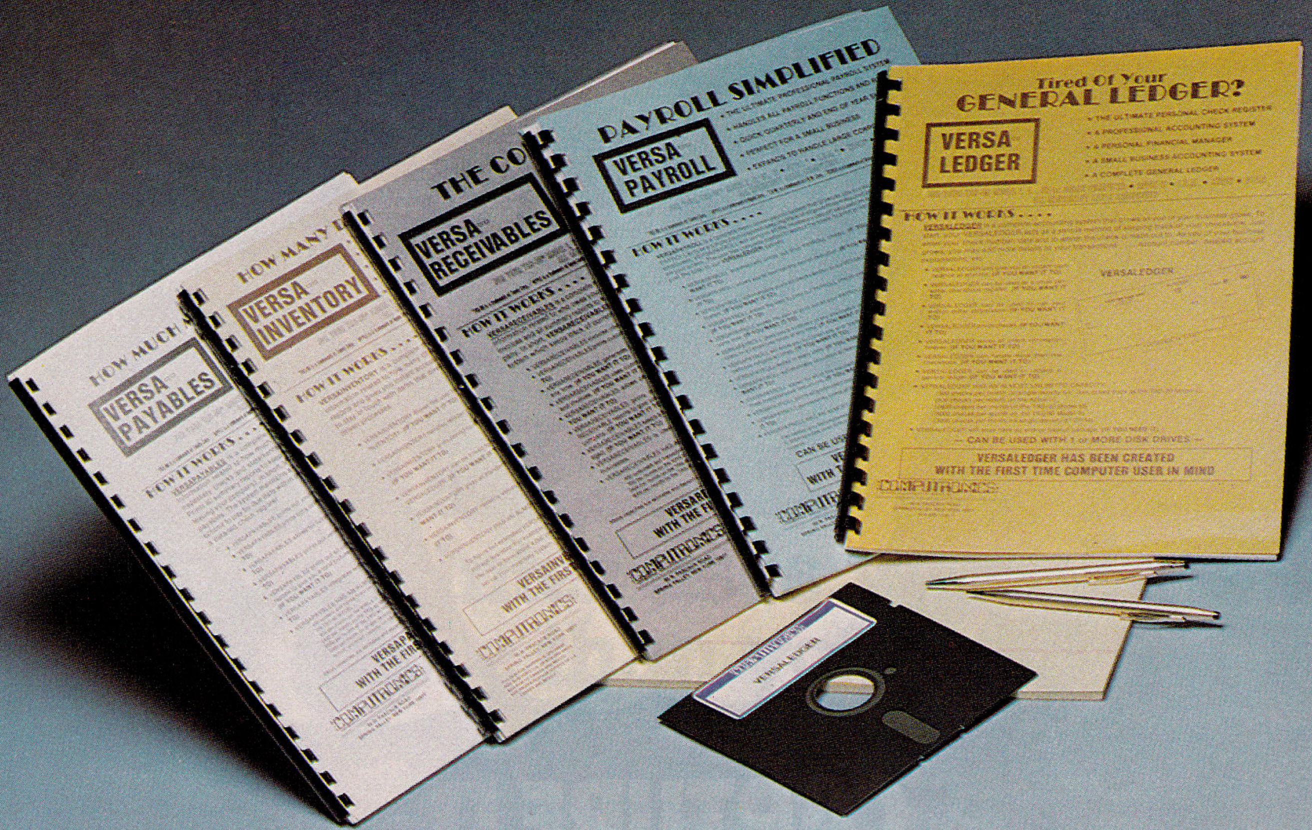
Keyboard buffer. Have you ever

Command	Pgm. Size	DOS 3.3	Diversi-DOS
LOAD	87 sectors	22 sec.	5 sec.
LOAD	30 sectors	9 sec.	3 sec.
SAVE	87 sectors	32 sec.	6 sec.
BLOAD	11 sectors	4 sec.	2 sec.

Table. Speeds of selected disk operations, using standard DOS and Diversi-DOS.

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • XEROX™



The VersaBusiness™ Series

Each **VersaBusiness** module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VersaReceivables™ $99.95

VersaReceivables™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VersaReceivables™ prints all necessary statements, invoices, and summary reports and can be linked with VersaLedger II™ and VersaInventory™.

VersaPayables™ $99.95

VersaPayables™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VersaPayables™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VersaPayables™, you can even let your computer automatically select which vouchers are to be paid.

VersaPayroll™ $99.95

VersaPayroll™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VersaLedger II™ system.

VersaInventory™ $99.95

VersaInventory™ is a complete inventory control system that gives you instant access to data on any item. VersaInventory™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VersaReceivables™ system. VersaInventory™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VersaLedger II™ $149.95

VersaLedger II™ is a complete accounting system that grows as your business grows. VersaLedger II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VersaLedger II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VersaLedger II™ comes with a professionally-written 160 page manual designed for first-time users. The VersaLedger II™ manual will help you become quickly familiar with VersaLedger II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every **VersaBusiness™** module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any **VersaBusiness™** module, you may return it within 30 days for a refund. Manuals for any **VersaBusiness™** module may be purchased for $25 each, credited toward a later purchase of that module.

To Order:

Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

DEALER INQUIRIES WELCOME

All prices and specifications subject to change / Delivery subject to availability.

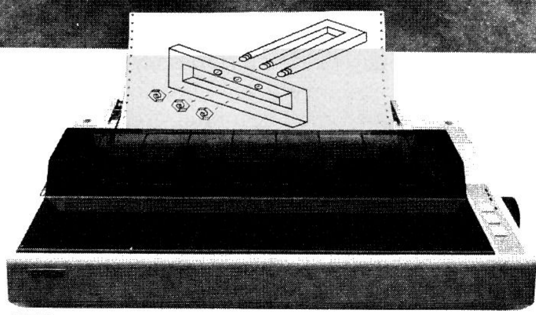
COMPUTRONICS INC.

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. - *APPLE is a trademark of Apple Corp. - *IBM is a trademark of IBM Corp. - *OSBORNE is a trademark of Osborne Corp. *CP/M is a trademark of Digital Research - *XEROX is a trademark of Xerox Corp.

Circle 66 on Reader Service card.





ACTUAL APPLE II PRINTOUT USING GRAPPLER AND EPSON MX100.

COMPLETELY REDESIGNED.
NOW, THE GRAPPLER + .

The original Grappler was the first graphics interface to give you hi-res screen dumps from your keyboard. The new Grappler + with *Dual Hi-Res Graphics* adds flexibility with a side-by-side graphics printout of page 1 and page 2.

The Grappler + can now be used with the Apple® Dot Matrix, the Okidata 84, and is Apple III compatible.* In addition, the IDS Grappler + is currently available with color capability, including color graphics screen dumps.

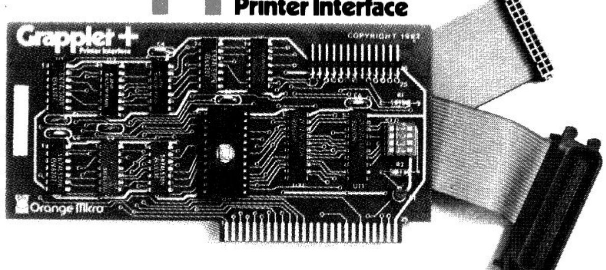
UP TO 64K BUFFER OPTION

An optional Bufferboard can now be added to all existing Grappler and Grappler + interfaces. See your Apple Dealer for details.

*Requires additional software driver.
**Requires graphics upgrade.

© Orange Micro, Inc. 1982

With The Grappler+™ Printer Interface



CPM is a registered trademark of Digital Research, Inc.
Apple is a registered trademark of Apple Computer, Inc.

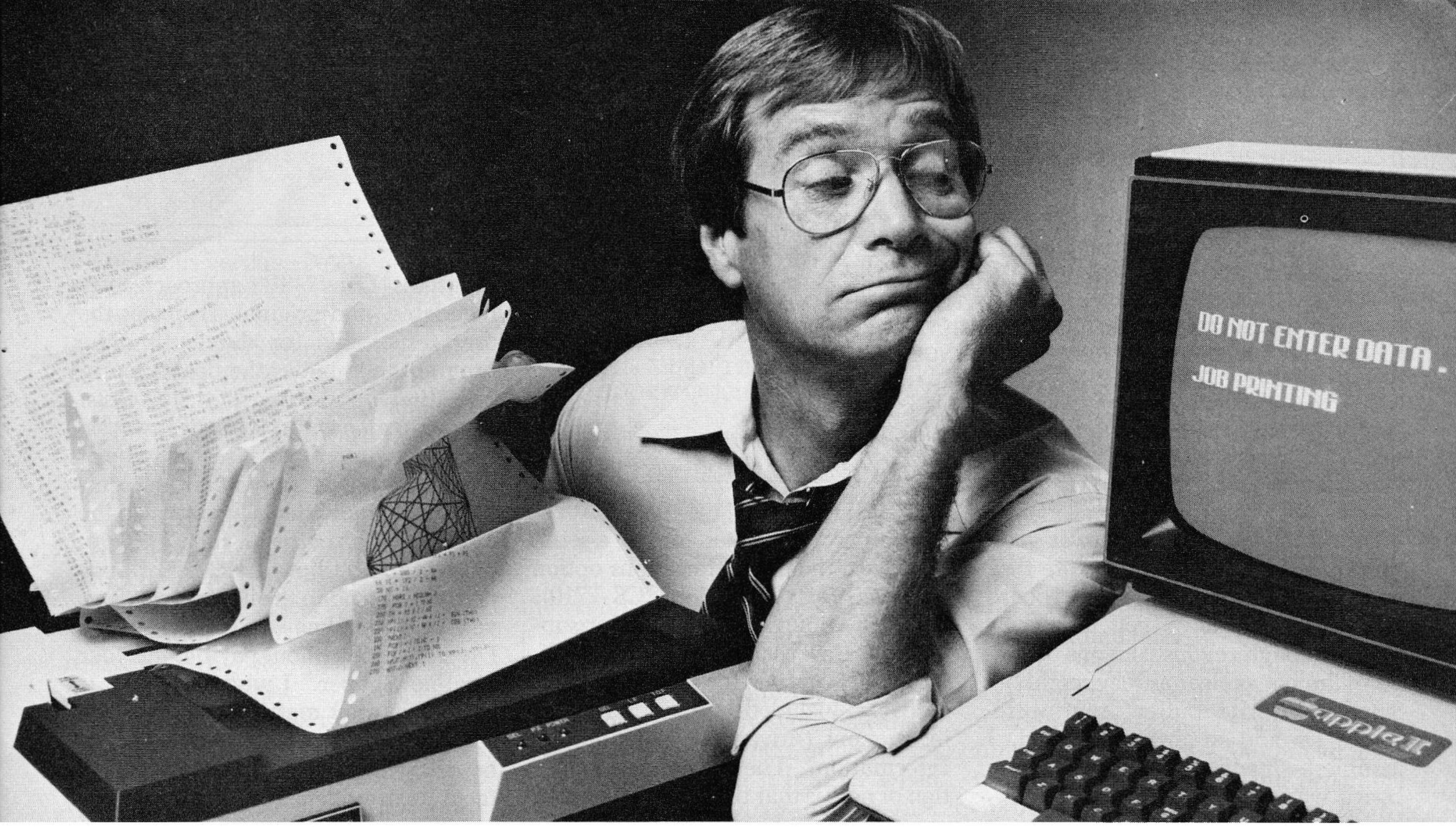
THE GRAPPLER + FEATURES:
• Dual Hi-Res Graphics • Printer Selector Dip Switch • Apple III Compatible* • Graphics Screen Dump • Inverse Graphics • Emphasized Graphics • Double Size Picture • 90° Rotation • Center Graphics • Chart Recorder Mode • Block Graphics • Bell Control • Skip-over-perf • Left and Right Margins • Variable Line Length • Text Screen Dumps • also works with Pascal and CPM.

THE GRAPPLER + INTERFACES WITH THE FOLLOWING PRINTERS:
• Anadex • Apple Dot Matrix • Centronics 122 • C. Itoh ProWriter • Epson MX-70, MX-80**, MX-80F/T**, MX-100 • IDS 460, 560, Prism 80 and 132, Microprism • NEC 8023 • Okidata 82A**, 83A**, 84.

Orange Micro inc.

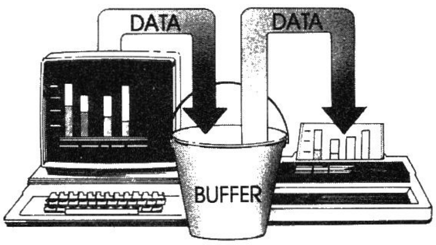
1400 N. Lakeview Ave.,
Anaheim, CA 92807 U.S.A.
(714) 779-2772 Telex: 183511 CSMA
Foreign Dealer Inquiries Welcome

Circle 5 on Reader Service card.



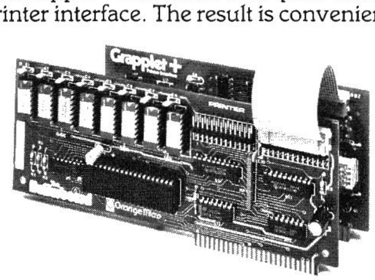
If your printer uses your Apple® more than you do, you need The Bufferboard.™

If your Apple is locked into the "PRINT" mode so much that you've taken up solitaire to kill the boredom, you need a buffer. And if your computer is the Apple II or III, the only buffer for you is The Bufferboard. Expandable to 64K of storage, The Bufferboard stores an instantaneous **bucketful** of print data from your computer. Then it feeds the data to your printer at its own printing rate. Your Apple is set free from driving your printer and is ready for more data from you.



Take your existing interface— and buffer it!

Only The Bufferboard has a simple Interface-Docking System. No bulky boxes or expensive power supplies are needed because The Bufferboard fits right into your Apple—and **docks** onto your existing printer interface. The result is convenient



and economical buffering of most popular printer interfaces, including the Grappler +™ interface, Epson interface, and Apple printer interface. Thirty seconds and a single hook-up are all you need to end the printer waiting game forever.

Up to 20 letter-size pages stored at a time.

The Bufferboard comes standard with 16K, and is expandable to 32K or 64K of buffering capacity with the addition of memory chips. This "bucket" will hold up to 20 pages of a print job, allowing you freedom to use your Apple.

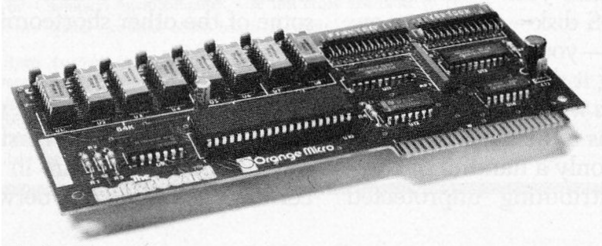
The Bufferboard—designed exclusively for the Apple Computer.

Specifications:
• Versions for Grappler + interface, Epson interface, Apple interface, and other popular printer interfaces • 16K buffer standard • Upgradeable to 32K or 64K • Automatic memory configuration • Automatic self test • Includes interface docking cable.

The Bufferboard is made by Orange Micro, Inc.; the same people who brought you the popular Grappler + printer interface. Both the Grappler + and The Bufferboard are now available at your local Apple dealer.

Apple is a registered trademark of Apple, Inc.
Epson is a registered trademark of Epson America, Inc.

Orange Micro inc.
1400 N. Lakeview, Anaheim, CA 92807
U.S.A. (714) 779-2772
TELEX: TX 183511 CSMA



The Bufferboard™
For Apples and Printers

```
DIVERSI-DOS(TM)
MASTER DISK - VERSION 1-A
COPYRIGHT 1982 DSR, INC
BY BILL BASHAM
1 — INSTRUCTIONS
2 — PUT DIVERSI-DOS ONTO AN UNPROTECTED DISK
3 — RUN A PROTECTED PROGRAM
4 — MAKE BACK-UP COPIES OF THE DIVERSI-DOS MASTER DISK
5 — MODIFY KEYBOARD/PRINT BUFFER ROUTINES
6 — ADD BUFFER ROUTINE TO ANOTHER DISK
7 — EXIT TO BASIC

ENTER SELECTION (1 TO 7) —>
```

Figure 1.
Menu for Diversi-DOS.

wanted to type ahead; to answer questions posed by a program before they are asked? Diversi-DOS's keyboard buffer routine allows you to do just that, although the instructions on this point are not as clear as they might be. Since the "print buffer" routine requires an add-on memory board, I assumed—incorrectly—that such a board was also necessary for the keyboard buffer. Instead, the buffer is handled within the RAM that comes with your Apple.

To activate the buffer, type BRUN BUFFER—assuming that you do own an add-on memory board. If you do not, it will be necessary to run menu option 5 first. When prompted, type N for DISABLE KEYBOARD BUFFER (Y/N) and Y for DISABLE PRINT BUFFER (Y/N). The new buffer routines will now be written to disk and can be accessed at any time.

The instructions mention a few limitations concerning the keyboard buffer. First, the programs using the buffer must use Get or Input statements to request data. Those which read the keyboard via Peeks will not work with the buffer installed. Also, when assembly-language programs are running, and during garbage collection, the routine may occasionally miss a character.

Print buffer. This feature reportedly sends the material to be printed to an add-on memory board rather than the printer. When the printer is ready, it pulls the data from the memory board and prints. With this feature, the Apple is free to perform other tasks while the printer is running. This routine will also make use of larger memory boards (up to 128K), but a 16K board is the required minimum. It was written for Saturn Systems' memory boards. If your board uses the same bank-switching scheme as the Saturn board does, DSR says it should work.

The print routine assumes that you have a parallel printer card or a serial card with the communications option (such as Versa-Card, or CCS 7710). The old Apple serial card is not supported. However, directions for modifying the routine for other types of cards are included in a section of the instructions (Advanced Programming Information). The only other limitation of note is that the print buffer routine disables the software features on the printer card. If your printer expects commands like control-I 80N, for example, to set the number of columns, you're out of luck. You will be restricted to 40-column output and standard printing—no overstrikes, italics or double-width.

"I was impressed with the features I was able to test. The DOS speed-up alone is worth the price."

Distribution Policy

I wanted to save this for last because DSR's policy is unusual, and definitely requires comment. The information shown in Figure 2 is presented each time you boot the Diversi-DOS disk. Thus, if you make your own copy of the Diversi-DOS disk—as you are encouraged to do—you save $5 over the cost of ordering it directly from DSR.

In these times when advanced copy-protection seems to be the watchword of the day and only a handful of companies are distributing unprotected programs, I find DSR's approach commendable, and original. It makes the user responsible for distributing the program, eliminates the dealer markup entirely, and cuts advertising costs. A winner all around? Only if it works. I'd be very interested in seeing some of the future sales receipts from companies and individuals who send in the $25 fee. As a matter of fact, DSR intends "to publish a partial list of businesses that send in the license fee. Conversely, if you know of someone using Diversi-DOS without paying, tell your local newspaper." I wish them luck.

As a related matter, DSR requires that all programs using Diversi-DOS be sold with the message (Figure 2) intact. I can understand why this is so, but I would hesitate to try to sell any of my programs this way. Picture the user who purchases a new program from XYZ Software and finds, upon booting the disk, that he owes another company $25 for using a utility that he didn't even know he was buying! This is the penalty attached to the "free, no royalty required" use of Diversi-DOS.

I would prefer including a message of credit to DSR and their address. If the user appreciated the feature, the complete package of routines (including the buffer program) could be ordered from DSR. The free advertising might easily be as successful as the "send $25" approach and, I feel certain, would be more likely to be included on disks or in program manuals.

Please don't assume that I did not like Diversi-DOS or find it useful. I was impressed with the features I was able to test. In my opinion, the DOS speed-up alone is worth the price. However, I feel, as does DSR, that the program limitations must be pointed out. The instruction set is complete in this regard and shows you how to correct such problems where possible. Since all the programs are unprotected, if you have any assembly-language skills, you may be able to overcome some of the other shortcomings.

Addendum

I recently received an examination copy of Diversi-DOS Version 1-B. A few brief comments are in order concerning the differences between it and

Figure 2.
Message appearing on Diversi-DOS.

DIVERSI-DOS(TM)

PLEASE COPY THIS DISK AND GIVE IT TO EVERYONE YOU KNOW!!!!!!!!!!!!!!!!!!!!

IT IS LEGAL TO COPY AND DISTRIBUTE DIVERSI-DOS, SUBJECT TO CERTAIN RESTRICTIONS (SEE "LICENSING INFORMATION"). HOWEVER, TO LEGALLY RUN THE DIVERSI-DOS PROGRAM ON THIS DISK, PLEASE SEND $25 DIRECTLY TO:

DSR, INC.
5848 CRAMPTON CT.
ROCKFORD, IL 61111

YOU WILL RECEIVE A VALIDATION STICKER AS PROOF OF PAYMENT. YOUR HONESTY WILL ALLOW US TO DISTRIBUTE FUTURE PROGRAMS THIS SAME, LOW-COST WAY. (ILLINOIS RESIDENTS ADD 5% SALES TAX)

the previous version (1-A).

First, according to Mr. Basham, 1-A did not function correctly in conjunction with the DOS command Append. This has been fixed in 1-B.

Second, when the buffer option is in effect, the printer card software commands are still disabled. However, the author has kindly pointed out in the revised instruction section that POKE 44764,44 does more than just turn off output to the screen when printing. In fact, it sets your printer to its full width, so that you will not be restricted to 40-column output, as was suggested previously.

Third, the new disk is currently being sent free to all licensed users of Diversi-DOS. Mr. Basham is also writing a program to move Diversi-DOS to a 16K RAM card. If you would like a copy of the program, send $5 and your User Support Number to DSR.

Fourth, there was a copy of an Epson screen dump program included on the examination disk I received. Since it will not operate on my Epson as a text dump and I do not have the graphics ROM, I assume that it is a Hi-Res dump. Note, however, that there is no mention of this new utility in the instructions nor does it transfer when making backup copies of the disk (menu option 4). As such, it is unclear whether this will be a part of Version 1-B or was intended simply as a sample of things to come.

Fifth, DSR and I apparently think alike concerning the probable hesitation of companies who make use of Diversi-DOS to include the "please send $25" message on their disks. DSR now offers the option of paying a one-time licensing fee of $200 instead and including a message in the program manual concerning Diversi-DOS availability.

DSR, by encouraging users to copy the programs, offers the opportunity to "try before you buy" and, if you like what you see, send in the license fee. ■

Practically Pascal Part 1

Might as well get in the mood for some hands-on work with Pascal, because it's implementation time. An uncomplicated and useful address book will be your reward. Just follow the step-by-step instructions—you'll find Pascal isn't as complex as you feared.

by John Stephenson

Here is your chance to contrast programming techniques, and, incidentally, produce an electronic address book using Pascal. Starting from the specification developed step by step in this article, try to implement it on the Apple II. Next month, compare the results.

Getting Started

The Apple Pascal operating system contains over 300K of code. Because the system is larger than the Apple's memory, only small chunks of it can reside in the Apple's RAM at any instant. Different parts are loaded as needed, overlaying other parts no longer being used.

The main command line's options are summarized in Table 1. Command lines indicate which part of the system is active and which options are available. Single alpha keystrokes invoke various parts of the system in a hierarchical manner. For example, pressing E for Edit summons the screen-oriented editor, which displays the edit command line. Valid keystrokes are set off from their option words by left parentheses. Returning to a higher level of system operation is usually accomplished by pressing Q for Quit.

Table 1. Main command line summary (Apple Pascal 1.1).

```
E(dit Invokes the editor.  The editor is used to type in programs,
      assembly language routines, and text documents, as well as save
      the files to diskette.

R(un Executes a program found in the file *system.wrk.code or summons
     the compiler to compile the file *system.wrk.text.

F(ile Invokes the filer.  The filer is used mainly to transfer and
      delete files, and display diskette directories.

C(omp Invokes the compiler.  The compiler translates program ".text"
      files into p-code ".code" files, which can subsequently be
      executed.
```

Table continued.

Vacillating between Edit and Compile is typical of the coding process, as illustrated in Figure 1. When a text file created with the editor compiles without generating error messages, then it is a syntactically consistent Pascal expression. The resultant code file may be executed to uncover run time errors. Both compile time errors (syntax) and run time errors (logic) are corrected by modifying the source text file using the editor.

The Process

The program development process consists of three essential phases: planning, coding and accepting. The complexity of these phases and the volume of documentation accompanying each phase varies with the size of the system to be developed, the number of professionals involved and the requirements of the end user. Even an elementary program written by one individual re-

Address correspondence to John Stephenson, 9118 Smith Ave., N. Bergen, NJ 07947.

Table continued.

L(ink Invokes the linker. The linker combines ".code" files produced by the assembler with ".code" files produced by the compiler into a single executable ".code" file. In other words, it combines 6502 native code with p-code.

X(ecute Begins interpretation of a ".code" file. The command is similar to RUN in basic.

A(ssem Invokes the assembler. The assembler is used to write routines in 6502 code. Common reasons to program in assembly language instead of Pascal are to handle specialized peripherals on the chip or register level, and to write time critical routines.

D(ebug Invokes the system debugger. In Apple Pascal 1.1 this feature has not been implemented.

? Pressing the "?" key displays the remaining part of the main prompt line, described below.

U(ser This restarts the last executed program. It is faster than repeating the X(ecute command because it bypasses certain otherwise necessary initialization procedures.

I(nitialize This checks for devices attached to the system and tries to execute a file called *SYSTEM.STARTUP.

H(alt Exits the p-system. Halting on AppleII+ machines causes a reboot.

S(wap Toggles the swapping option which defaults to off when the system is booted. When on, it maximizes the available ram for program execution at the expense of swapping in and out parts of the system module in use. For example, the editor can handle files up to 34 blocks long with swapping off, but up to 40 blocks long with swapping on. However, with swapping on, the editor must always be online.

M(ake exec Copies your commands to a diskette file. Later you may drive the system from the file instead of from the keyboard. Good for often repeated series of commands. Similar to basic EXEC files.

quires planning, coding and accepting. Coding without planning is a mistake.

The Plan

The planning phase consists of a definition and a design. The definition answers the "what" part of the problem. The design answers the "how" part of the problem, but in broad terms. If a program were being developed as contract work, then the client should be directly involved in the planning phase, ultimately approving a written specification. If a program is being developed as a product for the

"It is better to expose and remedy inconsistencies before the actual product is produced."

open market, then market research should be conducted, and, in fact, drive the plan. In both cases, a preliminary guide should be written and reviewed by potential users to ensure that the planned program is consistent with their needs and expectations. It is better to expose and remedy inconsistencies before the actual product is produced.

For program AddrBook, the programmer and the end user are one and the same, which simplifies the planning process. Because Program AddrBook is intended more as an exercise than an entry to the software marketplace, it is not essential to know what competing products are available or what their shortcomings might be. Therefore, the "what" part of the problem can be specified simply by designing the operating screen, emphasizing display style and functionality. The "how" part of the problem can be specified by writing the first layer of the program, which will define the logical flow.

Here is a simplified definition: Program AddrBook will provide a computerized mechanism for storage and retrieval of names, addresses and telephone numbers, somewhat analogous to common personal address books. The operating screen will look like Figure 2. The program employs a prompt line, similar to the usual Pascal system promptline, to manipulate the electronic address book. Below the promptline, status messages will be displayed. The remainder of the screen will display the address book entry currently being worked on. Each address book entry will consist of a name line, address line, telephone number line and note line, all of which can be entered freeform. You can add, delete, search, modify, list, print and clear entries. Entries will be maintained in alphabetical order keyed on the name line. The first character of the name line will always be an alpha. The entries in the address book will be

```
----->  E(dit   --------->   C(ompile   --------->   X(ecute   ------>  DONE
         ↑                      :                       :
         ↑                      :                       :
         ↑                      V                       :
         ↑ <----------- compiler errors                 :
         ↑                                              V
         ↑ <---------------------------------- run time errors
```

Figure 1.
Coding cycle.

Listing 1. Code shell.

```
program AddressBook;

{ PROGRAM TO MANIPULATE AN ELECTRONIC ADDRESS BOOK AND STORE IT ON DISK. }

{ GLOBAL DATA DEFINITIONS FOLLOW... }

const
        title          = 'PHONE BOOK [1.A]  ';
        maxList        = 100;
        fileName       = 'PhoneBook.Data';

        nameLength     = 30;
        addrLength     = 50;
        numberLength   = 15;
        noteLength     = 50;

        indent         =  5;
        displayAt      =  8;

type
        entryState     = (blankState,changedState,filedState);

        charSet        = set of char;

        phoneEntry     = packed record
                name   : string[nameLength];
                addr   : string[addrLength];
                number : string[numberLength];
                note   : string[noteLength];
                end;

        phoneRecord    = packed record case boolean of
                false  : (size:integer);
                true   : (persons:phoneEntry);
                end;

        phoneBook      = packed array [0..maxList] of phoneRecord;
var
        entryIsIn      : entryState;
        fileModified,
        quit           : boolean;
        command        : char;
        mainSet        : charSet;
        book           : file of phoneBook;
```

Listing continued.

stored and retrieved from disk in an efficient manner.

Here is preliminary user documentation: Bring up the Apple Pascal system as usual. Enter the filer and set the prefix to the disk name containing the file ADDRBOOK.CODE. Q(uit the filer and X(ecute ADDRBOOK. If this is your first time using the program, the message PHONEBOOK.DATA NOT FOUND. CREATE IT? Y/N will appear on the screen. Respond by pressing the Y key and an empty file will be initialized. After the data file is loaded or created, the program's main prompt line will be shown at the top of the screen.

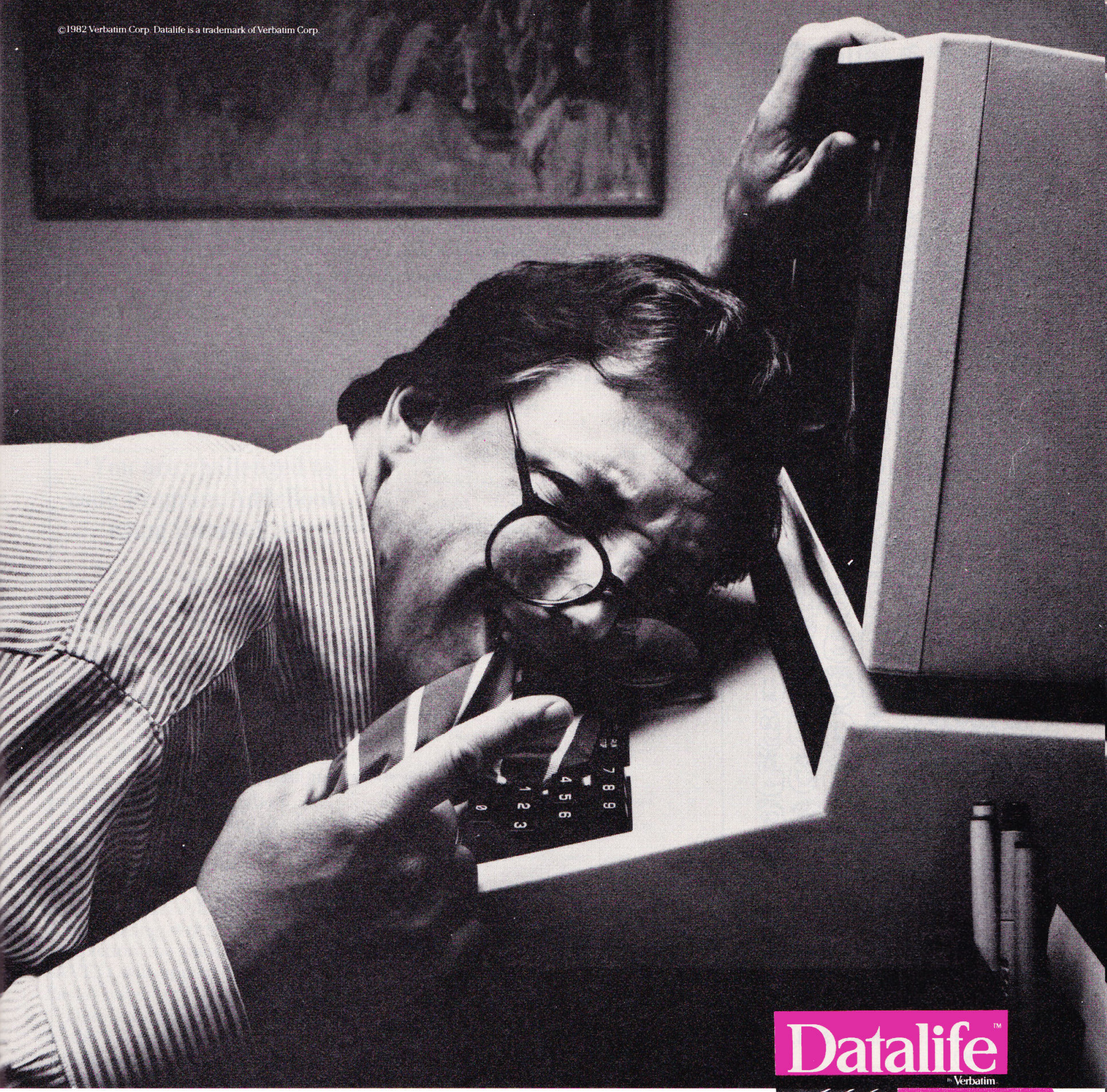
Below the prompt a message will inform you about how many entries are currently in the address book. A blank address form will be printed, indicating you are working on a blank entry.

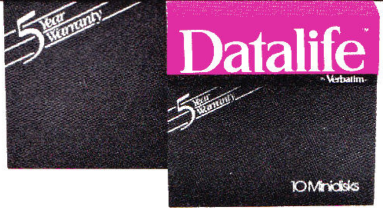
Select Add by pressing A if you wish to add a new entry. You will be prompted KEY ON WHAT NAME: . Type in the name. Enter the information and press the return key. You will probably want to type in the last name and then the first name since the information is maintained in alphabetical order. The name will automatically be inserted in the blank form. The cursor will be positioned on the address line, waiting for further input. Enter the address on the address line, the phone number on the phone line, and any additional information on the note line. The information will be automatically written in the address book. You will see the message RECORD ADDED.

Select Clear by pressing C to display another blank address book entry form.

Select Search by pressing S to locate and display an address book entry. In response to the prompt KEY ON WHAT NAME:, type a name. If an entry exactly matching the name you entered is located, it will be displayed; otherwise, the message NOT FOUND will be displayed.

Select List by pressing L to show the entries in the address book on the screen in alphabetical order. A new prompt line, "<space> for next item or Q quits," will be shown. If the end of the address book is reached, the message END OF BOOK REACHED will appear. Pressing the space bar again will wrap around to the first en-





How to avoid becoming a casualty of the computer revolution.

The computer revolution is eliminating lots of work for everybody. Unfortunately, it's often work you wanted to keep.

Because of inconsistent or weak magnetic properties, some flexible disks can "forget" some or all of your information. But not if you're using Datalife® flexible disks. They're certified 100% error free and backed by a 5-year warranty. This means what you put on them stays, ready for instant retrieval.

So enjoy all the benefits of the computer revolution without suffering the painful loss of your information. Use Datalife by Verbatim, the world's leading producer of flexible disks.

For your nearest Verbatim dealer, call toll-free 800-538-1793; in California or outside the U.S., call collect (408) 737-7771.

Circle 89 on Reader Service card.

Listing continued.

```
        currentEntry    : phoneEntry;
        currentIndex    : integer;

                        { PROCEDURE STUBS FOLLOW... }

function prompt (promptString:string; allowed:charSet) : char;
                { FUNCTION TO DISPLAY A PROMPT STRING AND RETURN
                  A CHARACTER AMONG THE ALLOWED SET }
begin
end;

procedure openFile;
                { PROCEDURE TO LOAD ADDRESS BOOK FROM DISK }
begin
end;

procedure saveFile;
                { PROCEDURE TO SAVE THE ADDRESS BOOK TO DISKETTE }
begin
end;

procedure clearEntry;
                { PROCEDURE TO FILL CURRENT ENTRY WITH NULLS }
begin
end;

procedure add;
                { PROCEDURE TO ADD A NEW ENTRY TO THE ADDRESS BOOK }
begin
end;

procedure delete;
                { PROCEDURE TO DELETE AN ADDRESS BOOK ENTRY }
begin
end;

procedure search;
                { PROCEDURE TO SEARCH FOR A RECORD KEYED ON LAST NAME }
begin
end;

procedure modify;
                { PROCEDURE TO MODIFY THE CONTENTS OF AN ADDRESS BOOK ENTRY}
begin
end;

procedure list;
                { PROCEDURE TO LIST CONTENTS OF ADDRESS BOOK ON CONSOLE }
begin
end;

procedure quitProgram;
                { PROCEDURE TO EXIT CLEANLY }
begin
end;
```

Listing continued.

try in the book. Press Q to return to the main prompt line.

Select Print by pressing P to copy all or part of the address book to the line printer. A new prompt line, "A-Z selects a section <space> all 0 quits," will be shown. Pressing a letter will print all names beginning with that letter along with their telephone numbers. Pressing space will print all names and telephone numbers in order. Pressing 0 returns to the main prompt line.

"You are challenged to code Program AddrBook using the shell."

Select Modify by pressing M to change the entry being displayed. A new prompt line, "E(ntire rec N(ame A(ddr P(hone R(emark Q(uit," will be shown. Press a key corresponding to the part of the entry you wish to modify. The cursor will be automatically placed in the proper position for you to type new information.

Select Quit by pressing Q when you are done updating and inspecting the address book. If the address book has been altered in any way, the message SAVING FILE will appear as the new data is being written to disk. Upon completion the message FILE NOW SAVED will appear followed by the Pascal system main command line.

Here is a simplified design: Listing 1 is the outer level of the program, which represents the transition between the functionality as expressed by the display screen and the actual code. The global declarations provide the necessary data to define the address book. The main program identifies what operations may be performed upon that data. The case statement turns the screen specifications into logical procedure calls. The procedures are represented as blank stubs to be filled in during the coding phase.

You are challenged to code Program AddrBook using the shell in the program listing, adding whatever support procedures you feel are necessary, and to compare it with the implementation that will appear in next month's continuation of this article. ■

Listing continued.

```
procedure print;
      { PROCEDURE TO SEND NAME AND NUMBER REPORT TO THE LINE PRINTER }
begin
end;

                    { MAIN PROGRAM }
begin
fileModified:=false;
currentIndex:=0;
quit:=false;
mainSet:=['A','C','D','L','M','P','Q','S'];
openFile;
repeat
   command:=
    prompt('A(dd C(lear D(elete L(ist M(odify P(rint Q(uit S(earch',mainSet);
   case command of
        'A': add;
        'C': clearEntry;
        'D': delete;
        'L': list;
        'M': modify;
        'P': print;
        'Q': quitProgram;
        'S': search;
   end {case};
until (quit);
if (fileModified) then saveFile;
end.
```

```
PHONEBOOK [version] A(dd C(lear D(elete L(ist M(odify P(rint Q(uit S(earch
[ message line ]



        --------------------------------------------------------

        Name:

        Address:

        Phone:

        Note:

        --------------------------------------------------------
```

Figure 2. Screen definition.

Space Empires

**Strategy deals the important hand in this game.
Meet the emergencies head on,
and perhaps you'll win a game or two.**

by Lee E. Sumner, Jr.



Magazines for other computers are an often overlooked source of programs for your Apple. Space Empires was written for the TRS-80 computer by James L. Smith, of Rogers, AR, and was published in the August 1981 issue of *80 Micro*.

I converted the program (made minimal changes), then I enlivened some of the screens with inverse characters and a few beeps that the TRS-80 was lacking. The only hard part was substituting the subroutines at line 2590 to compensate for the Apple's lack of PRINT USING. The following is taken from Smith's original article.

You are in charge of an exploration and colonization fleet that must colonize 19 star systems in 20 years (turns) to be considered a success. Each turn is divided into four phases.

Status Report

A status report begins each turn. On it are listed revenues (megacredits) earned for each system explored (5 megacredits) and each system colonized (15 megacredits). Also, any balance left from the previous turn is shown. The status report includes the number of systems you have colonized,

Listing. Space Empires conversion for Apple (from original printed in 80 Micro).

```
100  HOME : VTAB 5: HTAB 7: PRINT "* * * SPACE EMPIRES * * *"
110  HTAB 19: PRINT "BY"
120  HTAB 13: PRINT "JAMES L. SMITH"
130  PRINT : PRINT "THIS IS A GAME OF SPACE EXPLORATION.    PLAYERS MUST
      MOVE THEIR STAR FLEETS     CORRECTLY AND ECONOMIZE WISELY TO"
140  PRINT "COLONIZE ALL 20 STAR SYSTEMS AND WIN THEGAME.": PRINT : HTAB
     4: PRINT "APPLE TRANSLATION BY LEE SUMNER."
150  VTAB 20: PRINT "ARE YOU READY? HIT RETURN FOR THE FUN!";
160  GET U$
170  CLEAR
180  HOME :T = 2500
190 S$ = "C"
200  DIM G$(10,10)
210  VTAB 12: HTAB 5: PRINT "------ONE MOMENT PLEASE------"
220  PRINT : HTAB 5: PRINT "WE ARE CREATING THE UNIVERSE!"
230 G$(1,1) = "*"
240 C8 = 1:C9 = 1
250  FOR I = 1 TO 19
260 H9 =  INT ( RND (1) * 3 + 1):V9 =  INT ( RND (1) * 3 + 1)
270  IF H9 = 1 GOTO 310
280  IF H9 = 2 AND C8 < 9 THEN D8 = 2: GOTO 310
290  IF H9 = 3 AND C8 > 2 THEN D8 =  - 2: GOTO 310
300  GOTO 260
310  IF V9 = 1 GOTO 340
320  IF V9 = 2 AND C9 < 9 THEN D9 = 2: GOTO 340
330  IF V9 = 3 AND C9 > 2 THEN D9 =  - 2: GOTO 340
340 C8 = C8 + D8:C9 = C9 + D9
```

Listing continued.

Write to Lee Sumner, Jr., at 75 E. King St., Dallastown, PA 17313.

Listing continued.

```
350  IF G$(C8,C9) = "*" THEN D8 = 0:D9 = 0: GOTO 260
360  IF I = 19 THEN G$(C8,C9) = "C":I1 = C8:J1 = C9: GOTO 380
370 G$(C8,C9) = "*"
380 D8 = 0:D9 = 0
390  NEXT
400 T = T + 1:Y = 0
410  FOR I = 1 TO 10
420  FOR J = 1 TO 10
430  IF G$(I,J) = "." THEN 490
440  IF G$(I,J) = "*" THEN S1 = S1 + 1: GOTO 490
450  IF G$(I,J) = "E" THEN E1 = E1 + 1: GOTO 490
460  IF G$(I,J) = "C" THEN C1 = C1 + 1: GOTO 490
470  IF G$(I,J) = "S" THEN I1 = I:J1 = J: GOSUB 2260: GOTO 490
480  IF G$(I,J) = "R" THEN R1 = R1 + 1
490  NEXT J,I
500 ET = E1 * 5:CT = C1 * 15
510 G1 =  INT ( RND (1) * 20 + 1):G1 = G1 / 100 + .90
520 G2 =  INT ( RND (1) * 20 + 1):G2 = G2 / 100 + .90
530 ET = ET * G1:CT = CT * G2
540  HOME : HTAB 12: PRINT "CIRCA ";T;" A.D."
550  HTAB 8: PRINT "FEDERATION STATUS REPORT": PRINT : PRINT
560  PRINT  TAB( 3)"UNEXPLORED STAR SYSTEMS  ";S1
570  PRINT  TAB( 3)"SYSTEMS IN REVOLT        ";R1: PRINT
580  PRINT "REVENUE FROM ";E1;" EXPLORED SYSTEMS";
590 P = ET: GOSUB 2570:ET = P
600  GOSUB 2580: HTAB 37 - C: PRINT ET
610  PRINT "REVENUE FROM ";C1;" COLONY SYSTEMS  ";
620 P = CT: GOSUB 2570:CT = P
630  GOSUB 2580: HTAB 37 - C: PRINT CT
640  PRINT "BALANCE FORWARD";
650 P = TC: GOSUB 2570:TC = P: GOSUB 2580: HTAB 37 - C: PRINT TC
660 TC = TC + ET + CT
670  PRINT : PRINT "TOTAL FEDERATION REVENUE";
680 P = TC: GOSUB 2570:TC = P
690 P = TC: GOSUB 2580: HTAB 37 - C: PRINT TC: PRINT
700  PRINT  TAB( 5)"REPORT COMPLETE - HIT RETURN";
710  GET U$: HOME
720  IF X = 1 GOTO 1150
730  IF (TC + .005) < 7 THEN  PRINT "SORRY COMMANDER - YOU'RE SHORT OF F
     UNDS - NO MOVEMENT": GOTO 1150
740  PRINT "CIRCA: ";T;" A.D.";
750  PRINT  TAB( 22)"SPACE EMPIRES"
760  PRINT  TAB( 22)"MEGACREDITS";
770 P = TC: GOSUB 2570: HTAB 37 - C: PRINT TC: PRINT : PRINT : PRINT :Z =
     11
780  FOR J = 1 TO 10
790  PRINT  TAB( Z)J;
800 Z = Z + 2
810  NEXT
820 Z = 8:Y = 1
830  FOR I = 1 TO 10
840  PRINT : PRINT  TAB( Z)Y;
850 Z = Z + 1
860  FOR J = 1 TO 10
870 Z = Z + 2
880  IF G$(I,J) = "" THEN  PRINT  TAB( Z)".";: GOTO 900
890  PRINT  TAB( Z)G$(I,J);
900  NEXT J
910 Z = 8:Y = Y + 1
920  NEXT I
930  PRINT : PRINT : PRINT : PRINT : PRINT  TAB( 5)"FLEET MOVEMENT PHASE
      (V,H)";
940  INPUT V,H: HOME
950  IF V = I1 AND H = J1 THEN  PRINT "NO FLEET MOVEMENT THIS PHASE - AS
      ORDERED": GOTO 1150
960  IF V > 10 OR V < 1 THEN  PRINT "YOUR FLEET WAS ORDERED OUT OF THE U
     NIVERSE": GOTO 1800
970  IF H > 10 OR H < 1 THEN  PRINT "YOUR FLEET WAS ORDERED OUT OF THE U
     NIVERSE": GOTO 1800
980 TC = TC - 7
990  IF G$(V,H) = "" GOTO 1140
1000  IF V > I1 THEN I2 = V - I1: GOTO 1020
1010 I2 = I1 - V
1020  IF H > J1 THEN J2 = H - J1: GOTO 1040
1030 J2 = J1 - H
1040  IF I2 > 2 GOTO 1140
1050  IF J2 > 2 GOTO 1140
1060  IF S$ = "*" THEN S$ = "E"
1070 G$(I1,J1) = S$
1080 S$ = G$(V,H)
1090 G$(V,H) = "S"
1100  IF S$ = "E" OR S$ = "*" THEN E1 = E1 + 1
1110  IF S$ <   > "*" THEN  PRINT "YOU HAVE LANDED ON ONE OF YOUR SYSTEMS
      ": GOTO 1130
1120  PRINT "YOU HAVE SUCCESSFULLY EXPLORED ANOTHER  SYSTEM!"
1130  GOTO 1150
1140  PRINT : PRINT "SORRY COMMANDER. AN INCORRECT SET OF": PRINT "COORD
      INATES WAS ORDERED.  YOUR FLEET": PRINT "RETURNS TO PLANET FALL !"
1150  PRINT : IF X = 1 THEN  HOME : PRINT "YOUR FLEET IS STILL IN PORT":
      GOSUB 1860
1160  IF R1 > 0 THEN  GOSUB 2000
1170  IF  INT (T / 2) <   > T / 2 GOTO 1490
1180  PRINT : HTAB 5: FLASH : PRINT "* * * EMERGENCY SITUATION * * *": PRINT
      : NORMAL
1190 G4 =  INT ( RND (1) * 6 + 1)
1200  IF G4 = 1 AND X = 1 OR G4 = 2 AND X = 1 GOTO 1190
1210  IF G4 = 3 AND E1 = 0 AND C1 = 0 OR G4 = 4 AND E1 = 0 AND C1 = 0 GOTO
      1190
1220  IF G4 = 6 AND B3 = 1 OR G4 = 6 AND C1 <   = 1 GOTO 1190
1230  ON G4 GOTO 1240,1250,1270,1280,1290,1320
1240  PRINT "YOUR FLEET NEEDS FUEL AND SUPPLIES": GOSUB 1860: GOTO 1490
```

Listing continued.

explored, and systems remaining to be explored.

Movement Phase

A different universe (playing grid) is randomly created each game. On the first turn you begin with one colony where your fleet is based. Each turn you move to a new star system (*). It becomes an explored system (E). It will cost you seven megacredits each time you move. To move your fleet you must type the y-axis first, then a comma, then the x-axis and hit Return.

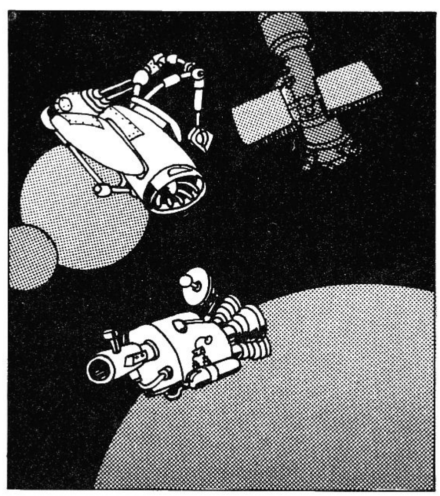
You can only move two spaces during a turn. You must land on a star system for a move to work. You cannot move outside the universe (playing grid). If you make a wrong move you are still charged seven megacredits, and your fleet returns to your last base. Plan ahead so you don't waste time by landing on systems you already own.

During the turns, various symbols will appear on the playing grid. C means a colony. E means an explored system. An asterisk * means an unexplored system. S shows the location of your ship.

Emergency Situations

On even numbered turns, you will have to meet and cope with emergency situations. The cost of meeting each emergency will be given. However, you may gamble and spend less to conserve revenue. If your gamble does not work, or you spend nothing, you must suffer the consequences. Possible emergencies are:

- Fleet repairs and supplies.



Listing continued.

```
1250  PRINT "YOUR FLEET IS IN NEED OF REPAIRS THIS   YEAR."
1260  GOSUB 1860: GOTO 1490
1270  PRINT "A STAR SYSTEM NEEDS MEDICAL AID": GOTO 1340
1280  INVERSE : HTAB 6: PRINT "A STAR SYSTEM IS IN REVOLT!": NORMAL : GOTO
     1340
1290  PRINT "THE WORLD COUNCIL DECREASES YOUR PRESENTALLOTMENT OF MEGACR
     EDITS BY 1/6 FOR THISYEAR."
1300 TC =  INT (TC - (TC * .16))
1310  PRINT "CURRENT BALANCE IS ";TC;" MEGACREDITS.": GOTO 1490
1320  PRINT "ONE OF YOUR COLONIES HAS BEEN DESTROYED BY A NATURAL DISAST
     ER.  IT HAS BEEN": PRINT "REDUCED TO EXPLORED STATUS."
1330  GOSUB 2500:B3 = B3 + 1: GOTO 1490
1340 R1 = R1 + 1
1350  IF E1 = 1 AND S$ = "E" OR E1 = 1 AND S$ = "*" THEN E1 = E1 - 1:S$ =
     "R": GOSUB 2000: GOTO 1490
1360  IF E1 = 0 GOTO 1420
1370  FOR I = 1 TO 10
1380  FOR J = 1 TO 10
1390  IF G$(I,J) = "E" THEN E1 = E1 - 1: GOTO 1480
1400  NEXT J,I
1410  STOP
1420  IF C1 = 1 AND S$ = "C" THEN S$ = "R":C1 = C1 - 1: GOSUB 2000: GOTO
     1490
1430  FOR I = 1 TO 10
1440  FOR J = 1 TO 10
1450  IF G$(I,J) = "C" THEN C1 = C1 - 1: GOTO 1480
1460  NEXT J,I
1470  STOP
1480 G$(I,J) = "R": GOSUB 2000
1490  PRINT : HTAB 5: INVERSE : PRINT "+ + + BUILD COLONIES + + +": NORMAL

1500  PRINT : PRINT "IT COSTS 90 MEGACREDITS TO BUILD A      COLONY."
1510  PRINT "YOU HAVE ";TC;" MEGACREDITS REMAINING."
1520  PRINT "YOU HAVE ";E1;" SYSTEMS TO COLONIZE."
1530  INPUT "HOW MANY COLONIES DO YOU WISH TO BUILD? ";B
1540  IF B = 0 GOTO 1700
1550 B1 = B * 90
1560  IF B1 > (TC + .005) THEN  PRINT "THAT EXCEEDS YOUR BUDGET": GOTO 1
     530
1570  IF B > E1 THEN  PRINT "YOU CAN ONLY BUILD ";E1;" COLONIES": GOTO 1
     530
1580 TC = TC - B1
1590  IF E1 = 1 AND S$ = "E" OR E1 = 1 AND S$ = "*" GOTO 1680
1600  FOR I = 10 TO 1 STEP  - 1
1610  FOR J = 10 TO 1 STEP  - 1
1620  IF G$(I,J) = "E" GOTO 1650
1630  NEXT J,I
1640  STOP
1650 G$(I,J) = "C":L1 = L1 + 1:C1 = C1 + 1:E1 = E1 - 1
1660  IF B > L1 GOTO 1590
1670  GOTO 1700
1680 S$ = "C":L1 = L1 + 1:C1 = C1 + 1:E1 = E1 - 1
1690  IF B > L1 GOTO 1590
1700 Z2 = 0: PRINT "NUMBER OF COLONIES BUILT:";L1
1710  IF C1 =  > 20 GOTO 1770
1720  IF T = 2520 GOTO 1800
1730  IF E1 = 0 AND C1 = 0 GOTO 1800
1740  IF TC < 0 GOTO 1800
1750 S1 = 0:E1 = 0:C1 = 0:R1 = 0:CT = 0:J2 = 0:B1 = 0:ET = 0:L1 = 0:I2 =
     0
1760  GOTO 400
1770  HOME : GOSUB 2300
1780  PRINT : PRINT "YOU WIN THE ASIMOV EXPLORATION AWARD FORCOLONIZING
     ALL SYSTEMS"
1790  GOTO 1830
1800  HOME : HTAB 15: PRINT "GAME OVER!": VTAB 4: PRINT "* * OUT OF 20 S
     YSTEMS YOU COLONIZED * *": HTAB 19: PRINT C1: HTAB 10: PRINT "* * S
     TAR SYSTEMS. * *"
1810  VTAB 10: PRINT "YOU DID NOT REACH YOUR GOAL!": VTAB 15: HTAB 12: INVERSE
     : PRINT " SORRY, YOU LOSE.": NORMAL
1820  PRINT : PRINT : IF C1 > 17 THEN  PRINT "BUT YOU CAME CLOSE!!"
1830  VTAB 23: INPUT "DO YOU WISH TO PLAY AGAIN (Y/N)";Q$
1840  IF Q$ = "Y" GOTO 170
1850  END
1860 G1 =  INT ( RND (1) * 20 + 1):G1 = (G1 / 100 + .90) * 20:AE =  INT
     (G1)
1870  PRINT "TO REACTIVATE YOUR FLEET YOU CAN SPEND  ";AE;" MEGACREDITS.
     "
1880  PRINT "SPENDING LESS WILL BE A GAMBLE."
1890  PRINT "YOU NOW HAVE ";TC;" MEGACREDITS."
1900  INPUT "HOW MUCH DO YOU WISH TO SPEND? ";P
1910  IF P = 0 GOTO 1970
1920  IF P > (TC + .005) THEN  PRINT "THAT EXCEEDS YOUR BUDGET.": GOTO 1
     900
1930 TC = TC - P
1940  IF P =  > AE GOTO 1980
1950 P1 = P / AE:G3 =  RND (1)
1960  IF G3 < P1 GOTO 1980
1970 X = 1: PRINT "SORRY, YOUR FLEET IS STILL DRY DOCKED.": GOTO 1990
1980 X = 0: PRINT "YOUR FLEET IS NOW OPERATIVE."
1990 P1 = 0:AE = 0: RETURN
2000  PRINT : PRINT "NUMBER OF SYSTEMS IN REVOLT: ";R1
2010 G2 =  INT ( RND (1) * 20 + 1):G2 = (G2 / 100 + .90) * 20:AE =  INT
     (G2)
2020  PRINT "YOU CAN SPEND UP TO ";AE;" MEGACREDITS": PRINT "ON EACH ONE
     ."
2030  PRINT "YOU MAY GAMBLE BY SPENDING LESS ON EACH ONE."
2040  PRINT "YOU HAVE ";TC;" MEGACREDITS REMAINING."
2050  PRINT "HOW MUCH DO YOU WISH TO SPEND ON ONE": INPUT "SYSTEM? ";P
2060  IF P = 0 GOTO 2120
2070  IF P > (TC + .005) THEN  PRINT "THAT EXCEEDS YOUR BUDGET.": GOTO 2
     050
```

Listing continued.

● Revolts. You do not receive revenue from systems in revolt. Colonies in revolt must be rebuilt to win.

● Budget cuts. The world council bureaucracy will cut your budget by 1/6 sometime during the game.

● Natural disasters. If a colony is destroyed, it reverts to explored status and must be rebuilt.

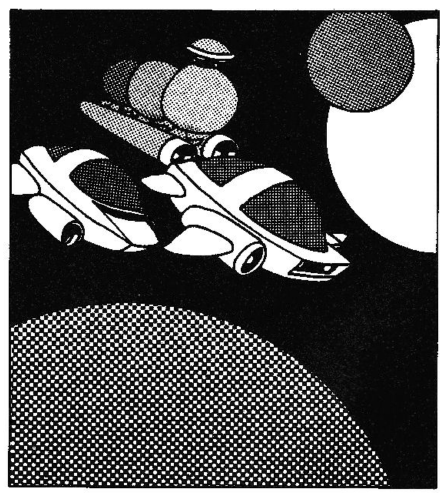
Build Colonies

You must build colonies as quickly as you can and pyramid your revenue to win the game. Only explored systems may be colonized. It costs you 90 megacredits to build a colony, so you won't be building right away.

Space Empires is never a sure win. You must plan your moves carefully, use some good strategy handling the emergency situations and have a little luck. ■

100–200	Prepare for the game
210–390	Create the Universe (grid)
400–530	Compute the revenue for present turn
540–710	Print the status report
720–940	Print the grid and call for movement
950–1150	Test movement accuracy and move fleet
1160–1480	Determine and print emergencies
1490–1700	Build colonies
1710–1760	Test for win/lose or return for next turn
1770–1790	Game win section
1800–1850	Game lose, play again section
1860–1990	Subroutine for fleet emergencies
2000–2250	Bring system out of revolt
2260–2290	Test for system where fleet landed
2300–2490	Winner display
2500–2560	Reduce colonies
2570–2630	Number formatting routines

Program structure table.



Listing continued.

```
2080 TC = TC - P
2090  IF P >  = AE GOTO 2130
2100 P1 = P / AE:G3 =  RND (1)
2110  IF G3 < P1 GOTO 2130
2120  PRINT "SORRY, YOUR SYSTEM IS STILL IN REVOLT.": GOTO 2250
2130  IF S$ = "R" THEN R1 = R1 - 1:S$ = "E":E1 = E1 + 1: GOTO 2210
2140  FOR I = 1 TO 10
2150  FOR J = 1 TO 10
2160  IF G$(I,J) = "R" GOTO 2200
2170  NEXT J
2180  NEXT I
2190  STOP
2200 R1 = R1 - 1:G$(I,J) = "E":E1 = E1 + 1
2210  PRINT "YOUR SYSTEM IS BACK TO NORMAL."
2220  IF R1 = 0 GOTO 2250
2230  INPUT "DO YOU WISH TO SAVE ANOTHER SYSTEM (Y/N)";Q$.
2240  IF Q$ = "Y" THEN P1 = 0: GOTO 2050
2250 P1 = 0:AE = 0: RETURN
2260  IF S$ = "C" THEN C1 = C1 + 1: GOTO 2290
2270  IF S$ = "R" THEN R1 = R1 + 1: GOTO 2290
2280 E1 = E1 + 1
2290  RETURN
2300  PRINT  TAB( 20)"#"
2310  PRINT  TAB( 20)"#"
2320  PRINT  TAB( 20)"#"
2330  PRINT  TAB( 20)"@"
2340  PRINT  TAB( 19)"@@@"
2350  PRINT  TAB( 18)"@@@@@"
2360  PRINT  TAB( 17)"@@@@@@@"
2370  PRINT  TAB( 16)"@@@@@@@@@"
2380  PRINT  TAB( 16)"@@@@@@@@@"
2390  PRINT  TAB( 16)"@@@@@@@@@"
2400  PRINT  TAB( 16)"@@@@@@@@@"
2410  PRINT  TAB( 16)"@@@@@@@@@"
2420  PRINT  TAB( 14)"@@@@@@@@@@@@@"
2430  PRINT  TAB( 13)"@@@  '''''  @@@"
2440  PRINT  TAB( 12)"@@@@  '''''  @@@@"
2450  PRINT  TAB( 11)"@@@@@  '''''  @@@@@"
2460  PRINT  TAB( 10)"@@@@@@  '''''  @@@@@@"
2470  PRINT  TAB( 9)"@@@@@@@  '''''  @@@@@@@"
2480  PRINT  TAB( 18)"'''''''"
2490  RETURN
2500  FOR I = 1 TO 10
2510  FOR J = 1 TO 10
2520  IF G$(I,J) = "C" GOTO 2550
2530  NEXT J,I
2540  STOP
2550 G$(I,J) = "E":C1 = C1 - 1:E1 = E1 + 1
2560  RETURN
2570 P =  INT (P * 100 + .05) / 100: RETURN
2580 Q = 1:P =  ABS (P)
2590 C = (P >  = 10) + (P >  = 100) + (P >  = 1000) + (P >  = 10000)
2600  IF P < 1 AND P > 0 THEN C = C - 1
2610  IF Q < 0 THEN C = C + 1
2620  RETURN
2630  REM
```

- Lines 1180, 1780 and 1800 should have some control-G beeps in the print statement. They do not show in the listing.
- I renumbered the program to make it easier to enter the program using some of the automatic line numbering facilities available for the Apple.

IIe Software Compatibility —A Report

Tooling up for the IIe doesn't mean scrapping your present software library. Here's a sampler of over 250 programs that make the upgrade.

by Hartley G. Lesser

It's great! The Apple IIe is just what you've been waiting for—64K bytes random access memory, enhanced keyboard with full ASCII character set, a redesigned back panel, new slot arrangements, 16K bytes of firmware in read-only memory (which includes a set of built-in self tests), new video handling routines, upper/lowercase, inexpensive 80-column board availability, an internal power-on light, easier servicing. . . but something seems to be worrying you. Could

Continued on page 80

Table 1. 2 indicates the program runs on a IIe with or without an 80-column card; 1 means it runs only without the 80-column card. N means incompatible.

A

A Stellar Trek	Rainbow Computing	strategy	2	* 3.2 DOS
A.E.	Broderbund Software	arcade	2	
A2-D31 Graphics Package	SubLogic Communications Corp.	graphics package	2	
A2-PB1 Night Mission Pinball	SubLogic Communications Corp.	pinball	2	
AS-SG4 Front Line	SubLogic Communications Corp.	strategy	2	
Adventure In Time	Phoenix Software	text adventure	2	
Air Navigation Trainer	Space-Time Associates	flight simulator	2	
Akalabeth	California Pacific	graphic adventure	2	
Alibi	Hayden Software	text adventure	2	
Alpha Plot	Beagle Brothers	graphics package	2	
Amper Sort/Merge	S&H Software	utilities	2	
Animator, The	BalbeSoftware Systems	animation package	2	
Apple Mechanic	Beagle Brothers	graphics utilities	2	
Apple Music Synthesizer (10-5-1)	ALF Products	music synthesis	2	
Apple Music Theory	Apple Computer	music education	2	
Apple Panic	Broderbund Software	arcade	2	
Apple Pascal Version 1.1	Apple Computer	Pascal language	2	
Apple Spice	Adventure International	Applesoft enhancer	2	
Apple Stellar Invaders	Apple Computer	arcade	2	* 3.2 DOS
Applesoft Program Optimizer	Sensible Software	utility	2	
Apventure To Atlantis	Synergistic Software	graphic adventure	2	
Arcade Machine, The	Broderbund Software	game creator	2	
Arithmetic Drill & Practice	Avant-Garde Creations	educational	2	
Artist, The	Sierra On-Line	graphics package	2	
Asteroid Belt	Superior Software	arcade	2	
Asteroids In Space	Quality Software	arcade	2	
Auto Graphics	Southwest Edpsych Services	graphics package	2	
Autobahn	Sirius Software	arcade	2	
Aztec	Datamost	animated adventure	2	

Table continued.

Table continued.

B				
Bandits	Sirius Software	arcade	2	
Banner Magic	Phoenix Software	print utility	2	
Bargain Box	Paul's Electric Computer	games	2	
Bargain Box II	Paul's Electric Computer	games	2	
Beagle Bag	Beagle Brothers	games	2	
Beer Run	Sirius Software	arcade	2	
Beneath Apple Manor (Special Edition)	Quality Software	fantasy adventure	2	
Birth of The Phoenix	Phoenix Software	text fantasy adventure	2	
Black Jack Strategy	Soft Images	blackjack tutorial	2	
Blade of Blackpool, The	Sirius Software	graphic fantasy adventure	2	
Bolo	Synergistic Software	arcade	2	
Bomb Alley	Strategic Simulations	strategy	2	
Bop-A-Bet	Sunnyside Soft	educational	2	
Bug Attack	Cavalier Computer	arcade	2	
Bumble Games	The Learning Company	educational	2	
Bumble Plot	The Learning Company	educational	2	
C				
Cannonball Blitz	Sierra On-Line	arcade	2	
Cartels and Cutthroats	Strategic Simulations	business simulation	N	
Casino	Datamost	gambling games	2	
Castle Wolfenstein	Muse Software	animated adventure	2	
Castles Of Darkness	The Logical Choice	animated adventure	2	
Caves Of Olympus	Howard W. Sams	graphic adventure	2	
Ceiling Zero	Turnkey Software	arcade	2	
Checkers	Odesta	checker game	2	
Chess	Odesta	chess game	2	
Choplifter!	Broderbund Software	arcade		* There are two versions. The latest version works on both II and IIe. Try first!
Combined Graphics Writer	Computer Stations	print utility	2	
Complete Graphics System II	Penguin Software	graphics package	2	
Computer Air Combat	Strategic Simulations	strategy/simulation		* 3.2 DOS
Computer Baseball	Strategic Simulations	baseball simulation	2	
Computer Bismark	Strategic Simulations	strategy		* 3.2 DOS
Computer Conflict	Strategic Simulations	strategy		* 3.2 DOS
Computer Foosball	Sirius Software	arcade	2	
Congo	Sentient Software	arcade	2	
Cosmic Balance, The	Strategic Simulations	strategy	2	
Cosmic Combat	Highlands Computer	arcade	2	
County Fair	Datamost	arcade	2	
Crazy Mazey	Datamost	arcade	2	
Creature Venture	Highlands Computer	graphic adventure	2	
Crime Stopper	Hayden Software	text adventure		
Crisis Mountain	Synergistic Software	arcade	2	
Crush, Crumble and Chomp!	Automated Simulations	arcade/strategy		* 3.2 DOS
Crystal Caverns	Hayden Software	text adventure	2	
Curse Of Crowley Manor	Adventure International	graphic adventure	2	
Cyber Strike	Sirius Software	arcade	2	
Cyborg	Sentient Software	text adventure	2	
Cytron Masters	Strategic Simulations	strategy	2	
D				
Dark Forest, The	Sirius Software	graphic adventure	2	
David's Midnight Magic	Broderbund Software	pinball		* There are two versions. The latest version works on both II and IIe. Try first!
Deadline	Infocom	text adventure	2	
Death Race 82	Avant-Garde Creations	arcade	2	
Diversi-DOS	Diversified Systems Research	DOS enhancer	2	
Dogfight	Microlab	arcade	2	
Doom Valley	Superior Software	text adventure	2	
DOS Boss	Beagle Brothers	utilities	2	
DOS Enhancer, The	S&H Software	DOS enhancer	2	
DOS Tool Kit	Apple Computer	assembler	2	
Dragon's Eye	Automated Simulations	graphic adventure		* 3.2 DOS
Dragon's Keep	Sunnyside Soft	graphic adventure (children)	2	
Dunjonquest: Temple of Apshai	Automated Simulations	graphic adventure		* 3.2 DOS

Table continued.

Table continued.

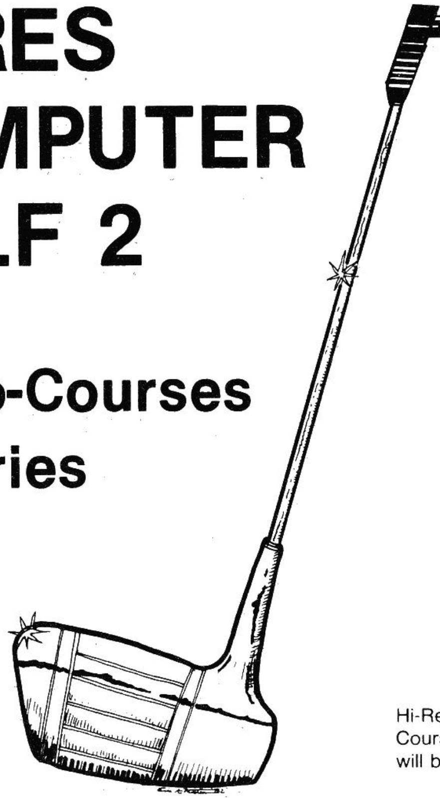
E				
Echo II Speech Synthesizer	Street Electronics	speech synthesis	2	
Electric Duet	inSoft	music synthesis	2	
Empire I: World Builders	Edu-Ware Services	graphic space adventure	2	
Empire II: Interstellar Sharks	Edu-Ware Services	graphic space adventure	2	
Epidemic!	Strategic Simulations	strategy	2	
Ernie's Quiz	Apple Computer	educational	2	
Expansion Packs I and II	Versacomputing	graphic aids (for Versawriter)	2	
F				
Federation	Avant-Garde Creations	arcade	2	
Final Conflict, The	Hayden Software	strategy	2	
Firebird	Gebelli Software	arcade	2	
Firefly	Muse Software	arcade	2	
Flex Text	Beagle Brothers	text utility	2	
Fonts, The	SoftSpoken	print utility	2	
Fracas	Quality Software	adventure		* 3.2 DOS
Frame-Up	Beagle Brothers	graphics utility	2	
Frazzle	Muse Software	arcade	2	
Free Fall	Sirius Software	arcade	2	
G				
Galactic Empire	Broderbund Software	strategy		* 3.2 DOS
Galactic Gladiators	Strategic Simulations	strategy	2	
Galactic Quest	Crystal Computing	strategy	2	
Galactic Revolution	Broderbund Software	strategy	2	
Galactic Trader	Broderbund Software	strategy	2	
Galactic Wars	Apple Computer	arcade/strategy	2	
Gamewriter	Creative Games International	game creator	2	
Genesis: The Adventure Creator	Hexcraft	adventure game creator	2	
Gold Rush	Sentient Software	arcade	2	
Golden Delicious Games For The Apple Computer	Wiley Professional Software	utilities	2	used in conjunction with tutorial book of same name
Gorgon	Sirius Software	arcade	2	
GraFORTH	inSoft	graphics language	2	
Gran Prix	Muse Software	arcade	2	
Graphics Application System	Avant-Garde Creations	graphics package	2	
Graphics Magician, The	Penguin Software	graphics package	2	
Graphics Processing System	Stoneware	graphics package	2	
Graphics Solution, The	Accent Software	graphics package	2	
Grapple	inSoft	arcade	2	
Guadal Canal Campaign	Strategic Simulations	strategy	2	
H				
Hi-Res Architectural Design	Avant-Garde Creations	graphics utility	2	
Hi-Res Electronics Design	Avant-Garde Creations	graphics utility	2	
Hi-Res Football	Sierra On-Line	sports game	2	
Hi-Res Secrets	Avant-Garde Creations	graphics package	2	
Hi-Res Soccer	Sierra On-Line	sports game	2	
House Of Usher	Crystal Computing	graphic adventure		* 3.2 DOS
I				
Ice Demons	Morningstar	arcade	2	
Illustrator, The	Island Graphics	graphics package	2	
Inspector, The	Omega Microware	utilities	2	
Instant Zoo	Apple Computer	educational	2	
Introductory Algebra	Avant-Garde Creations	educational	2	
Invasion Orion	Automated Simulations	strategy		* 3.2 DOS
J				
Jawbreaker	Sierra On-Line	arcade		* Two versions. The latest version runs on both the II and the IIe. Try first!
Joyport	Sirius Software	game port	2	
Juggle's Rainbow	The Learning Company	educational	2	
Juggler	Innovative Design Software	arcade	2	
L				
L.A. Land Monopoly	Continental Software	monopoly game		* 3.2 DOS
Labyrinth	Broderbund Software	arcade	2	

Table continued.

NEW RELEASES

HI-RES COMPUTER GOLF 2

Pro-Courses Series

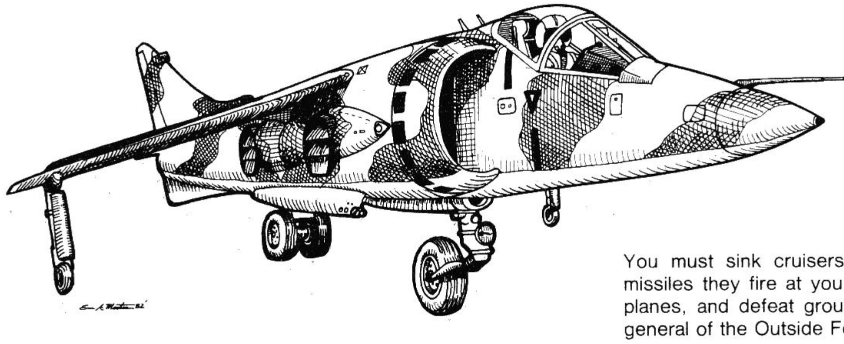


Spring is on its way, but why wait? You can be playing golf indoors now with Hi-Res Computer Golf 2, Pro Courses Series. This computerized version of one of America's most popular sports can be more of a challenge than the real thing! Introduced last year as Hi-Res Computer Golf, this improved version for 1983 requires even more skill and strategic planning than the original. Plus, you get all these great *new* features:

- Real professional golf courses presented in a multi-diskette system. The "Master" package and multiple "Pro-Courses" packages contain three professional courses each.
- Improved graphics for accurate reproduction of actual courses.
- Improved sounds.
- Auto-swing Option will take your swings for you but will allow you to override and take your own swing.
- Improved golf rules.
- Scorecard Archives Processor stores up to 20 complete or incomplete scores on each disk.
- Tournament-Security Option: Playing of nationwide tournaments is now possible thanks to a special feature which guarantees that the scores on a printed or handwritten scorecard have not been altered. Also prevents the golfer from re-playing a designated round.
- Plus many more improved features.

Hi-Res Computer Golf 2, Pro-Courses Series Master Disk Package with three Pro-Courses retails for $34.95. Additional Pro-Courses diskettes with three courses each will be available for $24.95.

JUMP JET



A Loyal island territory has been captured by a powerful army of outsiders. You must fly your Loyal Air Force JUMPJET on missions to regain the island. Your JUMPJET can outfly any aircraft in the sky, but you must deal with its limited fuel and weapons carrying capabilities.

You must sink cruisers and submarines while dodging bullets and missiles they fire at you, defend your carrier against kamikaze fighter planes, and defeat ground forces on the island. If you succeed, the general of the Outside Force will surrender.

JUMPJET is a fast-paced arcade-style game with state-of-the-art action and animation. It is bound to become one of your all time favorites. And it's available now for only $29.95!

These and many more exciting Avant-Garde Creations games are available now at your local dealer. Or you can order direct from us by mail or phone. Visa, MasterCard or American Express accepted. Order today!!

AVANT-GARDE CREATIONS **P.O. Box 30160** **Eugene, OR 97403** **(503) 345-3043**

AVANT-GARDE CREATIONS

Circle 52 on Reader Service card.

Table continued.

Labyrinth Of Crete	Adventure International	graphic adventure	2	
Laser Bounce	Hayden Software	arcade	2	
Lazermaze	Avant-Garde Creations	arcade	2	
Lisa	Sierra On-Line	assembler	2	
Lovers Or Strangers	Alpine Software	adult relationships	2	
LPS II	Gibson Laboratories	light pen	N	
Lunar Leeper	Sierra On-Line	arcade	N	
M				
Maces & Magic #2	Stone of Sisyphus	text adventure	N	
Maces & Magic #3	Morton's Fork	text adventure		* 3.2 DOS
Marauder	Sierra On-Line	arcade	2	
Mask Of The Sun, The	Ultrasoft	animated adventure		* Two versions. The latest version runs on both the II and IIe. Try first!
Menu Generator	Crane Software	utility	2	
Micro League Baseball	Stoneware	sports game	2	
Microwave	Cavalier Computer	arcade	2	
Minotaur	Sirius Software	arcade/adventure	2	
Missile Defense	Sierra On-Line	arcade	2	
Mix and Match	Apple Computer	educational	2	
Money Munchers	Datamost	arcade	2	
Mummy's Curse	Highland Computer	graphic adventure	2	
Music Maker	subLogic	music synthesis	2	
N				
Network	Edu-Ware Services	simulation		* 3.2 DOS
New Step-by-Step, The	Program Design	Basic tutorial	2	
Night Falls	Omega Microware	arcade	2	
O				
Odin	Odesta	Othello game	2	
Odyssey: The Compleat Apventure	Synergistic Software	graphic adventure		* 3.2 DOS
Olympic Decathlon	Microsoft Consumer Products	sports game	2	
OO-Topos	Sentient Software	text adventure	2	
P				
Paint Master	Avant-Garde Creations	graphics package	2	
Pandemonium	Soft Images	educational	2	
Pest Patrol	Sierra On-Line	arcade		* Two versions. The latest version runs on both the II and IIe. Try first!
Pie Man	Penguin Software	arcade	2	
Pig Pen	Datamost	arcade	2	
Pinball Construction Set	Budgeco	pinball game creator	2	
Pool 1.5	Innovative Design Software	pool game	2	
Poor Man's Graphics Tablet, The	Rainbow Computing	graphics package	2	
Pot O' Gold Plus	Rainbow Computing	assorted games	2	
Prisoner	Edu-Ware Services	adventure		* 3.2 DOS
Prisoner 2	Edu-Ware Services	graphic adventure	2	
Program Line Editor	Synergistic Software	utility	2	
Q				
Queen of Phobos, The	Phoenix Software	graphic adventure	2	
Quest for the Holy Grail	Superior Software	graphic adventure	2	
R				
Rapid Reader	Silicon Valley Systems	educational	2	
Red Alert	Broderbund Software	arcade	N	
Rendezvous	Edu-Ware Services	simulation	2	
Repton	Sirius Software	arcade	2	
Ribbit	Piccadilly Software	arcade	2	
Robot War	Muse Software	strategy	2	
S				
S.A.M.—The Software Automatic Mouth	Don't Ask Software	speech synthesis	2	
Sabotage	Sierra On-Line	arcade	2	
Sands of Mars	Crystal Computing	graphic adventure		* 3.2 DOS
Saturn Navigator	subLogic	simulation	2	

Table continued.

Table continued.

Sea Dragon	Adventure International	arcade with speech	2	
Seafox	Broderbund Software	arcade	2	
Sensible Speller	Sensible Software	speller	2	
Sentence Diagramming	Avant-Garde Creations	educational	2	
Serpentine	Broderbund Software	arcade	2	
Shapes In Color	Hayden Software	graphics package	2	
Shattered Alliance, The	Strategic Simulations	strategy	2	
Sheila	H.A.L. Labs	arcade	N	* compatible update near release
Sherwood Forest	Phoenix Software	graphic adventure	2	
Shuffleboard	Innovative Design Software	shuffleboard game	2	
Shuttle Intercept	Hayden Software	arcade	2	
Sight 'N' Sound	CompuGraphics Software	graphics & sound utility	2	
Snoggle	Broderbund Software	arcade	2	
Snooper Troopers— Granite Point Ghost	Spinnaker Software	educational	2	
Snooper Troopers— The Disappearing Dolphin	Spinnaker Software	educational	2	
Southern Command	Strategic Simulations	strategy	2	
Space	Edu-Ware Services	text adventure		* 3.2 DOS
Space Adventure	Sierra On-Line	graphic adventure	1	
Space Cadette	Funtastic	arcade	2	
Space Eggs	Sirius Software	arcade	2	
Space Raiders	United Software of America	arcade	2	
Space Vikings	subLogic	strategy	2	
Special Effects	Penguin Software	graphics utilities	2	
Speed/ASM	Sierra On-Line	machine language utility		* 3.2 DOS
Spitfire Simulator	Mind Systems Corporation	flight simulator	2	
Spotlight	Apple Computer	educational	2	
Spy's Demise	Penguin Software	arcade	2	
Star Blaster	Piccadilly Software	arcade	2	
Star Blazer	Broderbund Software	arcade	2	
Starship Commander	Voyager Software	strategy	2	
Sub Search and others	Beagle Brothers	game pack		* 3.2 DOS
Suicide	Piccadilly Software	arcade	2	
Super Disk Copy II	Sensible Software	utilities	2	
Super Starbase Gunner	Apple-Jack Software	arcade		* 3.2 DOS
Supertaxman 2	H.A.L. Labs	arcade	2	
T				
Tawala's Last Redoubt	Broderbund Software	graphic adventure	2	
Telengard	Avalon Hill Microcomputer	text adventure	2	
Teleport	Cavalier Computer	arcade	2	
Thief	Datamost	arcade	2	
Time Zone	Sierra On-Line	graphic adventure	2	
Track Attack	Broderbund Software	arcade	2	
Trick Shot	Innovative Design Software	pool game	2	
Tubeway	Datamost	arcade	N	
Tunnel Terror	Adventure International	arcade	2	
Type Attack	Sirius Software	educational/arcade	2	
Typefaces	Beagle Brothers	enhancement	2	
Typing Tutor II	Microsoft Consumer Products	educational	2	
U				
UCSD p-System, version IV	Softech Microsystems	operating system	2	
Ultima	California Pacific	graphic adventure	2	
Utility City	Beagle Brothers	utilities	2	
V				
V.C.	Avalon Hill Microcomputing	strategy	2	
Versawriter	Versa Computing	graphics tablet	2	
Vortex	Datamost	arcade	2	
W				
War	Adventure International	strategy	2	
Warp Destroyer	Piccadilly Software	arcade	2	
Warp Factor	Strategic Simulations	strategy	1	
Watson	Omega Software	utilities	2	
Wavy Navy	Sirius Software	arcade	2	

Table continued.

Table continued.

Wilderness Campaign	Synergistic Software	graphic adventure		* 3.2 DOS
Wizard and the Princess, The	Sierra On-Line	graphic adventure		* 3.2 DOS
Wizardry: Knight of Diamonds	Sir-Tech Software	graphic adventure	2	
Wizardry: Proving Grounds of Mad Overlord	Sir-Tech Software	graphic adventure	2	
Word Scrambler & Super Speller	Avant-Garde Creations	educational	2	
Z				
Zendar	subLogic	strategy	2	
Zero Gravity Pinball	Avant-Garde Creations	pinball	2	
Zoom Grafix	Phoenix Software	print utility	2	
Zork I	Infocom	text adventure		* 3.2 DOS
Miscellaneous				
13-Sector Boot Utility	Apple Computer	utility	2	
3.3 System Master Disk	Apple Computer	utility	2	

Continued from page 72

it be the price of the new IIe?

For starters, the basic computer with 64K RAM costs about $1395. There's a package price of around $1995 for a "bundle" arrangement: the basic 64K computer with an Apple III monitor, a single disk drive, and the Apple standard 80-column text card. Hmmm, not bad. Not much more than an equivalently equipped Apple II.

No, the price doesn't seem to be the stumbling block to upgrading your II. Then what's troubling you? The software? Ahhh, you have a large investment tied up in Apple II based software. What's the sense of buying an Apple IIe and then not being able to use those programs? Why bother upgrading your old Apple II if such is the case?

Take heart! After running numerous software programs through the Apple IIe, I found that most operate perfectly. The exceptions are programs that are autostart ROM dependant (for copy protection), many word processors (due to the Apple 80-column text card's method of accessing display information), and some of the older programs.

I tested all the listed software and hardware in the following manner. First, I booted the disks on the Apple II to ensure their working order, then on an Apple IIe with the new Apple standard 80-column text card in position.

On the IIe I used two methods—using the on/off switch and through the warm boot process. This latter method involves depressing the special open-apple function key first, and then the control and reset keys together. The control-reset keys are then released, with the one-apple key released last.

I ran each program through its paces. If the software failed to respond, I removed the 80-column text card from the computer, and tried again. In a few cases, removing the video card allowed the program to run normally. If this maneuver failed, I tried rebooting it on the Apple II to reassure myself that it was a functioning disk. If the disk booted, I once again tried the entire procedure on the Apple IIe.

Please keep in mind that only one Apple IIe was used to complete this study, and that the built-in memory tests were conducted before each disk boot. It's quite possible another unit might behave differently. This study was conducted in an attempt at fairness for both consumer and producing company.

I made no attempt to investigate all of the databases, word processors, and other more serious material. This is for a couple of reasons: I didn't have easy access to the software, or enough time to test every piece of software/hardware prior to this issue's deadline. Please note that future software and hardware reviews will include a comment on that particular item's compatibility with the Apple IIe.

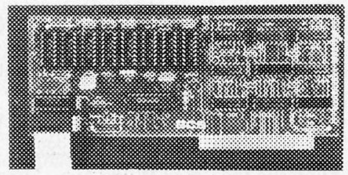
The software used for this study consists mainly of material from my

Table 2. Software suppliers.

Accent Software Inc., 3750 Wright Place, Palo Alto, CA 94306
Adventure International, Box 3435, Longwood, FL 32750
ALF Products Inc., 1448 Estes, Denver, CO 80215
Alpine Software, 2120 Academy Circle, Suite F, Colorado Springs, CO 80909
Apple Computer, 20525 Mariani Ave., Cupertino, CA 95014
Apple-Jack Software, P.O. Box 51, Cherry Valley, MA 01611
Automated Simulations, P.O. Box 4247, Mountain View, CA 94040
Avalon Hill Microcomputing Games, 4517 Harford Road, Baltimore, MD 21214
Avant-Garde Creations, P.O. Box 30160, Eugene, OR 97403
BalbeSoftware Systems, #6 White Plains, St. Louis, MO 63017
Beagle Brothers Software, 4315 Sierra Vista, San Diego, CA 92103
Broderbund Software Inc., 1938 Fourth St., San Rafael, CA 94901
Budgeco, 428 Pala Ave., Piedmont, CA 94611
California Pacific Computer Co., 1623 Fifth St., Davis, CA 95616
Cavalier Computer, P.O. Box 2032, Marina Del Rey, CA 92014
CompuGraphics Software, same as BalbeSoftware Systems

Table continued.

Apple II™ 16K RAM



49[95]

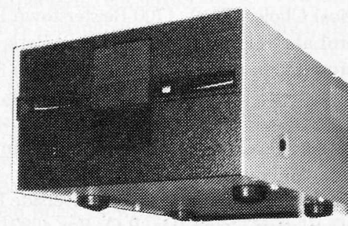
Compatible with Apple Language Card, Z-80 Softcard, Fortran, Pascal and CP/M. Assembled, tested and burned-in. With complete instructions and schematics. 90 day warranty.

C.B.E.

3375 Woodward Avenue
Santa Clara, California 95050
408/988-4408

We accept VISA and MasterCard. Dealer inquires invited.

225[00]



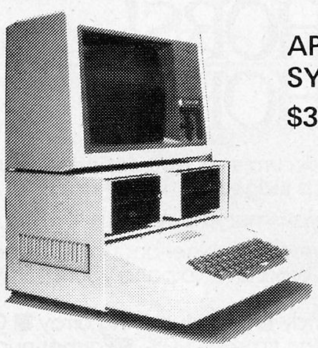
5¼″ APPLE COMPATIBLE DISK DRIVE

Shugart SA-390 Mechanism

Compatible with Apple Language Card, Z-80 Softcard, Fortran, Pascal and CP/M. Matching Apple color case and cable. 90 day warranty.

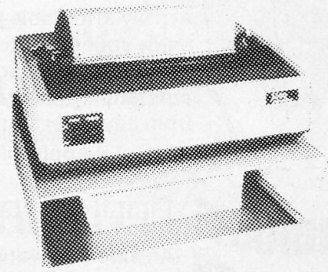
5¼″ FLOPPY BOX
$21.95 or 2/$37.50

As shown on our drive above.



APPLE SYSTEM SHELF
$37.50

All accessories are color-coordinated to Apple computers.



PRINTER PEDESTAL I
80 column – $24.95
PRINTER PEDESTAL II
132 column – $47.50

Amber or Green
COMPOSITE VIDEO MONITORS
HIGH-RESOLUTION – 20 Mhz

GREEN	9″	$129.00	12″	$139.95
AMBER	9″	$149.00	12″	$169.95

COLOR MONITORS

RGB and Composite Video inputs.
High-Resolution Graphics Capability.

Open Frame (OEM Style)
Enclosed Units
RGB-Host Adapters
Composite-Host Adapters

Call for prices!

Table continued.

Computer Stations Inc., 11610 Page Service Drive, St. Louis, MO 63141
Continental Software, 30448 Via Victoria, Rancho Palos Verdes, CA 90274
Crane Software, 16835 Algonquin, Suite 611, Huntington Beach, CA 92649
Crystal Computer, 12215 Murphy Ave., San Martin, CA 95046
Datamost, 9748 Cozycroft Ave., Chatsworth, CA 91311
Diversified Systems Research
Don't Ask Computer Software, 2265 Westwood Blvd., Los Angeles, CA 90064
Edu-Ware Services, P.O. Box 22222, Agoura, CA 91301
Funtastic Inc., 5-12 Wilde Ave., Drexel Hill, PA 19026
Gebelli Software, Sacramento, CA
Gibson Laboratories, 406 Orange Blossom, Irvine, CA 92714
H.A.L. Labs, 4074 Midland Road, Riverside, CA 92505
Hayden Software, 600 Suffolk St., Lowell, MA 01854
Hexcraft Inc., P.O. Box 39, Cambridge, MA 02238
Highlands Computer Services, 14422 S.E. 132nd St., Renton, WA 98056
Howard W. Sams, 4300 W. 62nd St., Indianapolis, IN 46268
Infocom, 55 Wheeler St., Cambridge, MA 02138
Innovative Design Software, P.O. Box 1658, Las Cruces, NM 88004
inSoft Inc., 10175 S.W. Barbur Blvd., Portland, OR 97219
Island Graphics, P.O. Box V, Bethel Island, CA 94511
Learning Company (The), 4370 Alpine Road, Portola Valley, CA 94025
Logical Choice (The), 1700 Reisterstown Road, Pikesville, MD 21208
MicroLab, 811 Stonegate, Highland Park, IL 60035
Microsoft Consumer Products, 400 108th Ave. N.E., Bellevue, WA 98004
Mind Systems Corporation, P.O. Box 506, Northampton, MA 01061
Morningstar, 39 Florence St., San Francisco, CA 94133
Muse Software, 347 N. Charles St., Baltimore, MD 21201
Odesta, 930 Pitner, Evanston, IL 60202
Omega Microware Inc., 222 S. Riverside Plaza, Chicago, IL 60606
Paul's Electric Computer, P.O. Box 42831, Las Vegas, NV 89116
Penguin Software, 830 4th Ave., Geneva, IL 60134

Table continued.

own software library, and material submitted to *inCider* for review.

The software is listed alphabetically. If you don't see a particular product on the list, that doesn't mean the material doesn't function on the Apple IIe, only that it wasn't tested.

You'll find all manner of product type, from entertainment to educational software, utility packages to graphics applications, in the listing of Table 1. Inclusion of a product is an indication of its compatibility. Following the product's name is the producing company. If a number 1 follows this information, then the software only operates on the IIe *without* the Apple standard 80-column text card installed. A number 2 indicates the software runs on the IIe with or without the same card installed. The letter N following a listing simply means the software is noncompatible. Additional explanations regarding use of the software on the IIe will follow the coded

result.

I think you'll agree that there seems to be little problem running Apple II software on the new Apple IIe computer. Just two short notes. First, for any software not specifically written for the Apple IIe that needs keyboard input (such as an adventure game), be certain to leave your IIe in Caps Lock mode. Otherwise, your entries/responses will be in lowercase, and lowercase is not recognized by Apple DOS (unless placed between quotation marks when coding a program). Second, any material that is indicated as * 3.2 DOS must either be muffined to 3.3 DOS, or booted after using the Basics Utility to reconfigure your DOS to 13-sector. You'll note that some programs have two versions available. Usually, the later version is compatible with both the II and the IIe. If the software isn't marked IIe compatible, I recommend you try booting the product on a IIe prior to purchase. ■

Table 2 continued.

Phoenix Software Inc., 64 Lake Zurich Drive, Lake Zurich, IL 60047
Piccadilly Software, 89 Summit Ave., Summit, NJ 07901
Program Design Inc., 11 Idar Court, Greenwich, CT 06830
Quality Design Software, 6660 Reseda Blvd., Reseda, CA 91335
Rainbow Computer Inc., 19517 Business Center Drive, Northridge, CA 91324
S&H Software, 58 Van Orden Road, Harrington Park, NJ 07640
Sensible Software, 6619 Perham Drive, West Bloomfield, MI 48033
Sentient Software, P.O. Box 4929, Aspen, CO 81612
Sierra On-Line Systems, 36575 Mudge Ranch Road, Coarsegold, CA 93614
Sierra Software, 536 East Sahara Ave., Las Vegas, NV 89104
Silicon Valley Systems, 1625 El Camino Real #4, Belmont, CA 94002
Sir-Tech Software Inc., 6 Main St., Ogdensburg, NY 13669
Sirius Software Inc., 10364 Rockingham Drive, Sacramento, CA 95827
Soft Images, 200 Route 17, Mahwah, NJ 07430
SofTech Microsystems, 9494 Black Mountain Road, San Diego, CA 92126
SoftSpoken, P.O. Box 7000-863, Redondo Beach, CA 90277
Southwest Edpsych Services, P.O. Box 1870, Phoenix, AZ 85001
Space-Time Associates, 20-39 Country Club Drive, Manchester, NH 03102
Spinnaker Software Corporation, 215 First St., Cambridge, MA 02142
Stoneware Incorporated, 50 Belvedere St., San Rafael, CA 94901
Strategic Simulations, 465 Fairchild, Mountain View, CA 94043
Street Electronics, 1140 Mark Ave., Carpenteria, CA 93103
SubLogic Communications Corp., 201 W. Springfield Ave., Champaign, IL 61820
Sunnyside Soft, 5815 East Parkside, Fresno, CA 93727
Superior Software Inc., P.O. Box 261, Kenner, LA 70065
Synergistic Software, 5221 120th Ave. S.E., Bellevue, WA 98006
Turnkey Software, 13078 Mindanao Way, Marina Del Rey, CA 90291
Ultrasoft Inc., 24001 S.E. 103rd St., Issaquah, WA 98027
United Software of America, 750 3rd Ave., New York, NY 10017
Versa Computing Inc., 3541 Old Conejo Road, Newbury Park, CA 91320
Voyager Software, P.O. Box 15-518, San Francisco, CA 94115
Wiley Professional Software, 605 3rd Ave., New York, NY 10158

Charting the Future with pfs:GRAPH

Display your data graphically with this special-purpose software from a famous maker.

by Steven C. Ross

Software Publishing Corp. has designed a software package to simplify the production of visual aids. The pfs:GRAPH package will produce charts on the Apple II screen, on the Hewlett-Packard 7470A Plotter, and on Silentype, Epson and other printers via the Grappler interface. Data for the graphs may be input by the user, or read from VisiCalc files or from other pfs: program files.

The package includes a manual, a program disk and a sampler disk containing data files to illustrate the various options. The sampler disk may be copied for backup, but the program disk may not be copied. A "Spare Copy Certificate" is included, however, and the purchaser may buy a backup copy of the program disk for $15. The program disk has no write-protect cutout, which eliminates the possibility of accidentally writing on it.

The manual is well written and covers the information in a logical sequence. The user is first given a brief introduction to the concepts behind the package, and then led through a demonstration of the various options. Following the demonstration are six chapters which treat each of the program functions in detail. I will discuss the operation of the program first, and then provide a critique.

Operation

To begin, turn the computer off, place the printer interface card (if any) in slot #1, and the plotter interface card (if any) in slot #2. Insert the pfs: GRAPH program disk in drive 1, and the sampler disk in drive 2. You can use a one-drive system, and the program will prompt you as necessary to insert the program or sampler disk. When all is ready, turn on the system and load the program. After about 20 seconds, the menu in Figure 1 will appear. This menu controls movement among the functions, which are related as shown in Figure 2.

Function 1, Get/Edit Data, is used to directly enter data, to read data from a VisiCalc .DIF file, or to read data from a pfs: file. Once data is entered, this function is used for editing. The manual is fairly clear as to how to get data from VisiCalc and pfs: files, and I was able to get data easily from both types of files by following directions in the manual. Entry of data from the keyboard is also covered in the manual. I will discuss a brief direct-data-entry example that illustrates a number of the concepts underlying the program.

Data is stored in *charts*, with each chart containing up to four *graphs*. There are two kinds of data, *X Data*, which may be identifier (alpha), numeric or date (days, months, quarters, years), and *Y Data*, which must be numeric.

In a bar graph, the X Data will appear across the horizontal axis, while the Y Data will be on the vertical axis. X Data values are constant across each of the four graphs, while the Y Data may differ. X Data is entered first; Y Data is then entered.

For example, you might want to graph forecast and actual sales for the past seven years. The X Data format would by Y(ears), and the values would be 1976, 1977,. . . 1982. Begin-

1 GET/EDIT DATA	4 SAVE CHART
2 DISPLAY CHART	5 GET/REMOVE CHART
3 DEFINE CHART	6 PRINT/PLOT

Figure 1. Menu for pfs:GRAPH functions.

Steven Ross is a professor in the College of Business Administration, Marquette University, Milwaukee, WI 53233.

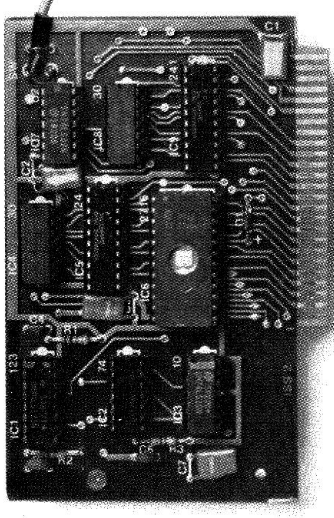
Circle 215 on Reader Service card.

NEW FOR APPLE II

WILDCARD™

MAKES BACK-UP COPIES OF PROTECTED SOFTWARE QUICKLY, EASILY, WITH JUST A PUSH OF A BUTTON.

New software locking schemes have rendered even the latest generation of copy programs virtually unusable. Locksmith™, Nibbles Away™ and other "Nibble copiers" require complicated parameter settings, much patience and great effort to use. More often than not, the results are disappointing. WILDCARD is different. Rather than copying disks track by track, WILDCARD ignores the disk and any copy protection encrypted on it. Instead, WILDCARD takes a snapshot of memory in your Apple® II. Now you can make back-up copies of protected software with the push of a button.



Features

- □ Hardware copying device... push button operation.
- □ Copies 48K memory resident software, most 64K software.
- □ No programming experience or parameters necessary.
- □ Backs up DOS 3.2 and DOS 3.3 disks.
- □ Creates DOS 3.3 unprotected and autobooting disks.
- □ WILDCARD lives in any slot. Undetectable by software.
- □ Produces autobooting disk in 2 minutes.
- □ Copies are DOS 3.3 compatible.
- □ Copies become accessible for alterations.
- □ Simple, easy-to-use software included.

WILDCARD ***Utility Disk 1*** also available, featuring:

- □ Automatic program compression and BRUN file maker.
- □ Multiple programs can be placed on the same disk.
- □ Recreates basic files to load and save.
- □ Files can be placed on a hard disk.....and more.

Software is not copy protected. System requirements: Apple II Plus with 64K and DOS 3.3 or Apple *IIe*. Franklin Ace also supported.
*Wildcard does not operate with CP/M® or other microprocessor based software.

Order direct from East Side Software Co., 344 E. 63 St., Suite 14-A, New York City 10021, 212/355-2860. Please include $3.00 for handling. Mail and phone orders may be charged to MasterCard and VISA. N.Y. State residents add sales tax. Dealer inquiries welcome.

WILDCARD	**$129.95**
WILDCARD *Utility Disk 1*	**$30.00**

IMPORTANT NOTICE: The WILDCARD is offered for the purpose of enabling you to make archival copies only. Under the Copyright Law you, as the owner of a copy of a computer program, are entitled to make a new copy for archival purposes only and the WILDCARD will enable you to do so. The WILDCARD is offered for no other purpose and you are not permitted to utilize it for any other use, other than that specified.

Apple II is a registered trademark of Apple Computer, Inc. CP/M is a registered trademark of Digital Research, Inc. Locksmith—trademark of Omega Microware, Inc. Nibbles Away—trademark of Computer: applications.

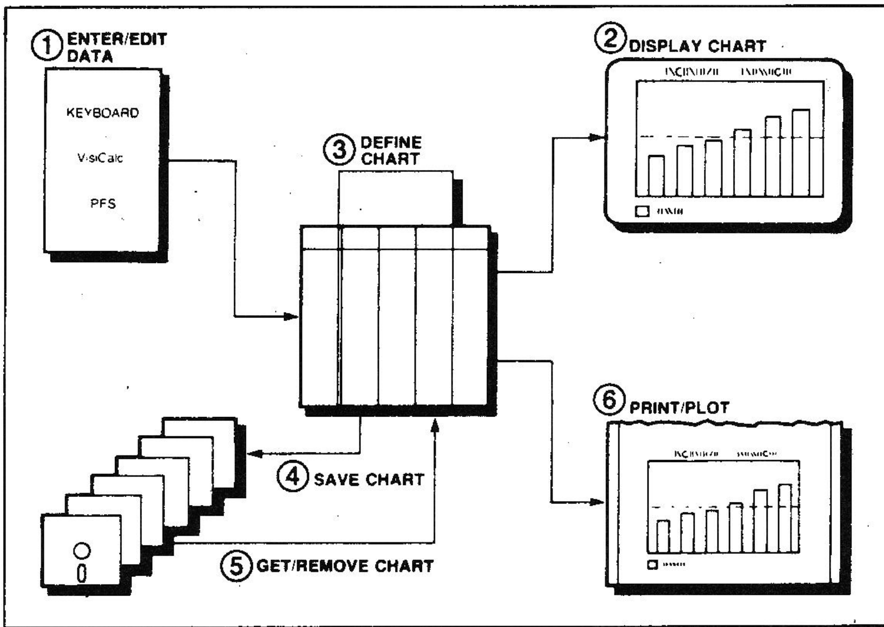


Figure 2.
Relationship between pfs:GRAPH functions. (Courtesy Software Publishing Corp.)

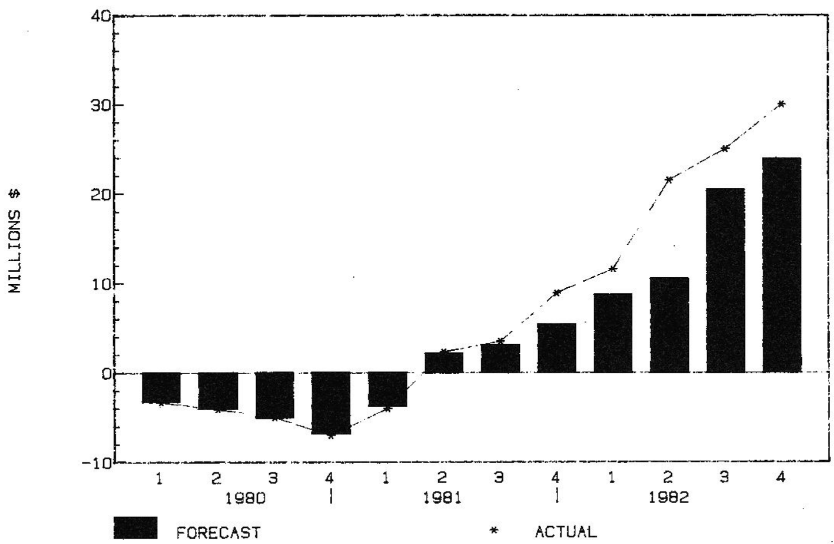


Figure 3. Bar and line combination.

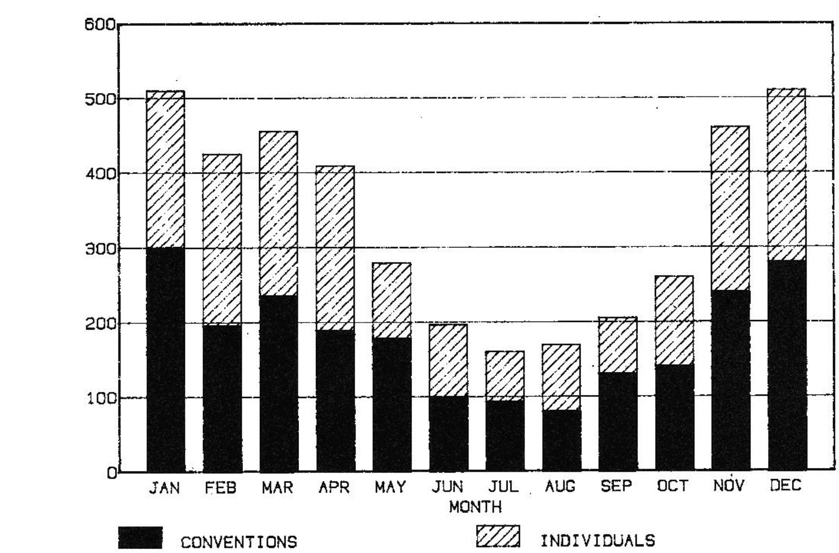


Figure 4. Stacked bars.

ning with graph A, each X Data point and a corresponding Y Data point would be entered.

In our example, you would enter the year, and then enter forecast sales for that year. After entering years and forecast sales, you would then enter the actual sales data in the Y Data column of graph B.

Once data is entered and edited, you may use function 2 to Display Chart. The chart is drawn on the Apple's screen in a very brief time (less than one minute in most cases), and you can see what the chart looks like.

The program has a number of default parameters for the chart, which may be modified using function 3, Define Chart. Within this function, the graphs may be labeled, the chart title and axis titles may be specified, and several options concerning the data display may be chosen. The graph type may be specified as *line*, *bar* or *pie*. The same chart may contain both line and bar graphs for an interesting effect (see Figure 3).

Minimum and maximum values for the Y axis may be set, as well as the number of divisions. Horizontal grid lines may be displayed or eliminated. The graphs may be stacked (see Figure 4) or arranged side by side (Figure 5). The chart can be in color, for a color monitor and the HP plotter. Noncolor charts use solid fill, hatching and cross-hatching to differentiate the graphs, as illustrated in Figures 2 and 3.

Function 4, Save Chart, is used to save the chart to a disk file. Once you ask for function 4, the program will display a list of all charts currently on the disk. You can direct the program to save the chart with a new name, or use an existing name to overwrite an earlier chart.

Function 5, Get/Remove Chart, will either retrieve a chart from disk or delete a chart on the disk. If you attempt to remove a chart, either explicitly with function 5 or implicitly with function 4, the program will stop to verify that you want to erase the chart.

To obtain a paper copy of the chart, use function 6. After selecting function 6, the program asks whether you wish to print or plot. The only *plotter* supported is the Hewlett-Packard 7470A. *Printers* supported include the Silen-

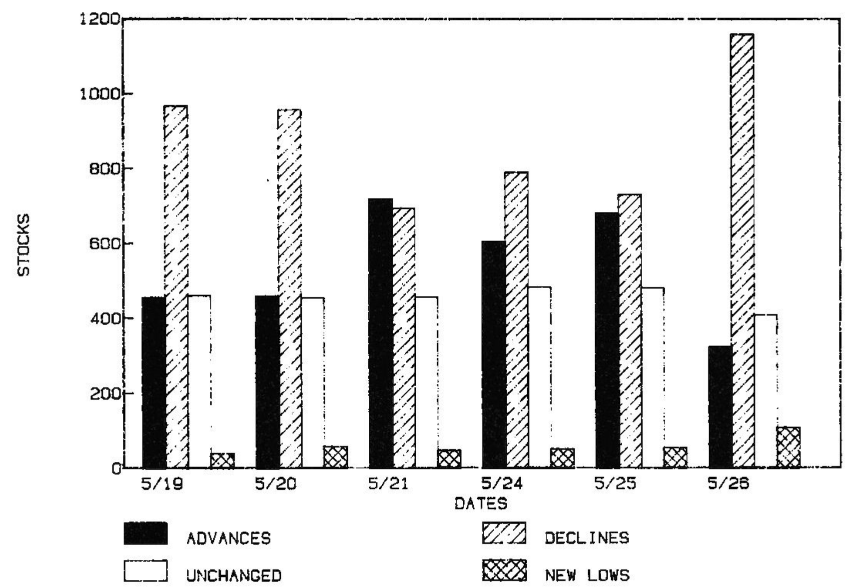
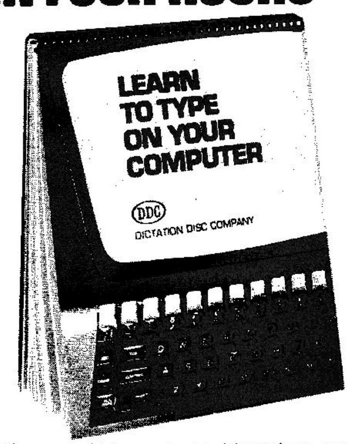


Figure 5. Unstacked bars.



type, the Epson series, and any printer connected with the appropriate Grappler interface card.

To print, you need specify only the type of printer, whether or not you wish the chart in expanded size, and whether or not the data should be printed. Figure 6 illustrates a nonexpanded chart printed, with data, by my Epson printer. This option is good for a quick view of what the finished copy will look like, and also is a convenient method for storing a copy of the data in a chart. The expanded size printer output fills a letter-sized sheet, and is suitable for many applications.

You can use the plotter to create a more impressive display, using either transparency or paper. You can specify whether or not the plotter should stop for you to change the pens. With no pen change, the plotter will plot all marginal information with one pen, and all graph information (regardless of color or no-color choice in function 3) with the other pen. If you ask the plotter to pause for pen change, you can change the pen prior to each graph being drawn. Multicolor charts are thus possible in either the color or the no-color mode. The chart will fill a letter-size sheet of paper unless you reset (manually, on the plotter) the reference points to make the chart smaller.

Critique

I found the pfs:GRAPH package easy to operate. The demonstration in the introductory chapter both illustrates the various options and gives the user confidence in his/her ability to successfully operate the program. You can create finished charts on the HP Plotter with minimal reading of the manual. The manual is well indexed for later reference.

The sampler disk contains a number of sample charts. A few of those charts are discussed in the manual, but others are not. I recommend that the publishers include an index of these samples, with an explanation of what each illustrates. Such an index (perhaps with a copy of each chart) would expedite the familiarization process because the user would be exposed to the variety of good examples.

I found a number of features of the program annoying, however. First,

"It is easy to use, as long as you want to do what it allows you to do."

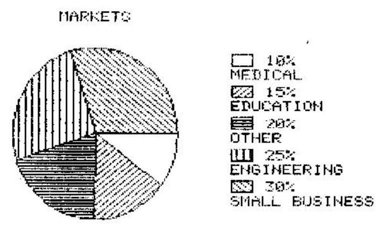


Figure 6. Printer example.

the program requires that the plotter interface card be in slot #2. This conflicts with those who use that slot for other functions, e.g., the Hayes Micromodem in conjunction with CP/M. I would think that the program could have been written to allow for optional slots. As it is now, I must disassemble my computer whenever I want to use the program to create plotter output.

The amount of data which may be displayed is rather limited. No more than 36 X Data points may be used (and less than 36 in some modes). No more than four sets of Y Data may be used. Thus, no more than 144 data points may be charted. I have found a number of instances where the capacity of the program was less than what I wanted.

Data entry can also be tedious. Perhaps most users of the program will work from pfs: or VisiCalc files, but those who have other filing systems must manually enter data. The procedure for entering data is fairly simple, but could be improved. First, you must specify the X Data format. If you enter more than one set of Y Data (more than one graph) then the X data format may not be changed except by erasing the entire chart and starting over. Because pie charts require I (alpha) type X Data, I found myself reentering data several times to get the appropriate type. The program also does not offer any ability to transpose the data matrix (i.e., exchange rows for columns). The result of these constraints is that the same data must be entered twice for applications which require both pie and bar charts. VisiCalc users will be faced with a similar problem: the data will have to be stored in two different .DIF files, once by rows and once by columns.

The charts produced on the HP plotter are good, although a few changes to the program would improve them. When the outline of the chart is drawn, the plotter repeats the line twice, yielding a heavier line. When lines are drawn within the chart, however, only one pass is made. These lines are faint. Better lines would result if the plotter were directed to draw each segment two or three times, with a slight offset in pen position to yield a bolder line. A second enhancement which could be easily built into the program would be the ability to scale the plot size under the control of the computer, rather than using somewhat inexact manual scaling.

Summary

I am moderately positive concerning this package. It is fairly constrained in what it allows, but it does what it does very well. I predict that some people will be extremely pleased with the program, while others will reject it as inadequate for their needs. For those who already have the pfs: family of software, and for those who use VisiCalc, this program provides an excellent capability for visual display of data. It is easy to use, as long as you want to do what it allows you to do. The package is also appropriate for those who want to make charts or transparencies for *ad hoc* presentations.

With a little practice, you can sit down and compose a chart on the screen, then dump it to printer or plotter and have the result in a fraction of the time and expense of having a professional graphics department do the work. However, I would advise those who do not use pfs: or VisiCalc to spend some time with the program to determine whether or not they are willing to cope with its data entry system to obtain the output. ■

dBASE II

Expect To Be Amazed!

Learning dBase II is similar to learning a new programming language, but the power and flexibility to be gained are, without question, worth the effort.

by Gregory R. Glau

Amazing! That's really the only word that comes to mind in reading through the dBASE II manual. I found myself thinking, "Can this program *really* do all these things?"

dBASE II is a product of Ashton-Tate, 9929 West Jefferson Blvd., Culver City, CA 90230; (213) 204-5570. To run it your Apple needs a Z-80 microprocessor, CP/M and 64K memory. The suggested retail price is $700.

But I'm getting a bit ahead of myself. What is dBASE II? Taken at face value the name suggests a database program, but, in fact, it's really much more. It can be almost anything you want it to be and can do just about anything you want it to do, when it comes to data handling. But you have to work at it—mastering dBASE II is like learning a new programming language.

Ashton-Tate's literature describes their creation as an "assembly-language relational database management system." They call it "an information handler, not a file handler," and "a tool that allows easy manipulation of small and medium sized databases using English-like commands." It also promises that you'll "gain a large measure of program/data independence, so that when you change your data you don't have to change your programs, and vice versa."

The literature further claims, "Users (say) they've cut the amount of code they write (to accomplish a specific task) by up to 80% with dBASE II."

Integrated Data

The manual notes that normal business accounting systems are broken down into distinct functions. The payroll data is processed by a payroll program, inventory is handled by an inventory program, and so on. These files are unrelated in both content and in the manner in which their information is handled.

If you'd like to get a feel for the relationship between the number of 10-inch Wilder shears in stock and the dollars you spent on advertising for those shears, coupled with estimated orders for the third quarter of 1983, you might spend half a day digging out the information and certainly a bit more figuring out how to comprehend it. dBASE II (from the manual again) "integrates the data and makes it much easier to get useful information from your records."

As alluded to above, dBASE II cannot take raw data and simply do what you want it to—not fresh out of the box. Instead, you must create the application programs. In this sense, perhaps dBASE II should be classified as a programming language, like Basic. In fact, you do use the commands from dBASE II to create a program. Although these programs don't use line numbers, they do operate sequentially and have characteristics similar to other high-level programming languages.

Convenient Features

dBASE II has a number of convenient features. You probably use random access files for your accounting system, perhaps saving 400 bytes (characters) for each file's information. A new program to be connected to the accounting system has to be aware of that fact. A line that saves information in a random access file using an incorrect file size will scatter your data hither and yon. dBASE II automatically takes care of discrepancies in file size. It can handle up to 1000 characters of information per record. It also has a simple method for increasing or decreasing the fields in each record. In Basic, on the other hand, if you leave room for those 400 characters of information in each file, that's all you get. If you've left something out, you have to write another program to transfer the data.

The dBASE II procedure to change file parameters is easy, too. You first

Greg Glau also writes inCider's *"Bent on Business" column. Address correspondence to Gregory R. Glau, PO Box 1627, Prescott, AZ 86302.*

```
ERASE
1> ENTER BILLS AND TIME SHEETS
2> PAY BILLS AND SALARIES
3> DEPOSITS AND CHECKBOOK
              Pick a number (Q to Quit)
WAIT TO Action
IF ! (Action) = 'Q'
     Quit to '&q'
ELSE
  IF Action = '1'
  DO CostMenu
        ELSE
          IF Action = '2'
           DO PayMenu
                ELSE
                  IF Action = '3'
                  DO DepMenu
                  WAIT
           ENDIF 3
        ENDIF 2
     ENDIF 1
  ENDIF Q
ENDO Accounting
```

Listing. dBASE II Command file.

create a new temporary file with the same structure as your original file, but without its data. You then make your modifications on this new file. After that, you bring the data in from the old file. The process is quick, easy, and to-the-point, especially when you consider the Basic code that would be required to perform the same function.

Further Comparisons To Basic

While dBASE II commands are similar to those of Basic, they have their own special meanings and structure. In Basic the normal way to do program branching is a GOTO or a GOSUB statement. dBASE II Command files, on the other hand, could be considered similar to the Apple EXEC command: they "order up" other programs from the main menu. Once they've done their work, control is RETURNed to the main menu once again.

Sample Listing

Listing 1 is an example. You'll note there are no program lines, and each menu selection calls up a new little program, much the same as a GOTO in Basic sends the program here and there. The indentations are to make the program easier to read.

The dBASE II Command file looks odd because it's unfamiliar, but it's pretty easy to follow the flow. It clears the screen (ERASE) and displays the menu, just as in Basic. Then it WAITs for you to enter what you want it to do.

Another difference from Basic is that each IF statement must be followed by an ENDIF, whereas a GOSUB in Basic requires a RETURN. Also, you can answer Q, a letter, to Quit the dBASE II program, even though it asked you to pick a number. While in Basic you have to use a string variable to accommodate both numbers and letters, in dBASE II you can assign alphanumeric data to what the Basic programmer would consider a numeric variable.

Creating a Database

It's important to get a feel for how dBASE II "thinks," especially in relation to another language, such as Basic. Consider for a moment how



Figure 1.
File structure printout.

```
STRUCTURE FOR FILE:  MAIL:LABEL.DBF
NUMBER OF RECORDS: 00103
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
FLD   NAME          TYPE    WIDTH    DEC
001   NAME            C       030
002   ADDRESS         C       025
003   CITY            C       020
004   STATE           C       002
005   ZIP:CODE        C       010
**total**                    00087
```

you'd create a simple mailing list database for your business in Basic. You would first estimate the number of characters each field would require. You'd assign a variable name to each field, and write programs to create a blank file on your disk and to enter the information. You'd also need a program to correct the data, and one to retrieve and print it. If you changed from the original label format, you'd have to redo the last program to adjust the printing accordingly.

How about dBASE II? Its file creation process is almost too easy to be true. You type CREATE. dBASE II responds:

ENTER FILE NAME:

You then enter the name you want to use. Say you're starting that mailing-label system; you might enter:

MAIL:LABEL.

dBASE II then prompts:

ENTER RECORD STRUCTURE AS FOLLOWS:

FIELD NAME, TYPE, WIDTH, DECIMAL
001

NAME is the name of the first field, probably a customer's name. There are three TYPEs of data: character (any character), numeric (number characters only), and logical (true and false). WIDTH refers to the number of spaces required for each entry. DECIMAL refers to whether the entry will have trailing numerals.

So, in answer to the first of these prompts, you type:

CUST:NAME,C,25

This tells dBASE II that the first field will be called CUST:NAME; will include letter characters, no more than 25 in number; and will have no trailing numerals.

You can enter data immediately into this new file. You can see what structure your file has, by typing LIST STRUCTURE. The printout for your mailing label file might look like Figure 1.

A listing of all the items in the database appears when you type DISPLAY ALL. Want to see just the first record? DISPLAY Record 1 will do it. How about the next 3? Type just that—DISPLAY Next 3.

GOTO TOP takes you to the first record in your file, GOTO BOTTOM, to the last. If you'd just like to poke around a bit, BROWSE displays up to 19 records and as many fields as will fit on the screen.

You can INSERT information where you want it. Forgot where you stopped entering data? More information to put into the system? Type USE MAIL:LABEL (to use the file you want), then write APPEND and a blank record will come up for your use.

By the way, when you want to enter data, field lengths are indicated on the screen by initial and terminal colons. The blank space between the colons is white (inverse).

Editing

Once your file has been created and information entered, you can alter it with full-screen editing. Editing with dBASE II is done using the control key in combination with various characters.

dBASE II also has a Replace function that allows you to adjust similar data all at once. For example, if you had an inventory file, with associated price lists, you could instruct the program to change all the prices at once. You can use REPLACE with conditions, asking it to adjust only those prices lower than a specified figure, or to change all prices higher than a certain price by a certain percent.

Data Retrieval

The raison d'etre of any information system is to get data out of it, and at

```
1.  USE MAIL:LABEL

    LIST for ZIP:CODE <= 70000

    ...this will display any file entries with a
       ZIP:CODE lower than or equal to 70000.

    Robert Rogers, 123 Maple, Santa Fe, NM 69137

2.  USE MAIL:LABEL

    DISPLAY ALL FOR NAME <'F'

    ...this will display all file entries who have
       a name starting with a letter 'lower' in the
       alphabet than 'F'.

    William Burroughs, 177 South Elm, Tucson, AZ 85719
    Marjorie Calendar, 230 Westbrook, Pima, AZ 86220

3.  USE MAIL:LABEL

    LIST FOR 'EE' $ NAME

    ...this will display all entries whose name
       name includes 'EE' in it.

    Harold Green, 20 West Parmette, Mayer, AZ 86321
    Rob Teening, 114 North MtVernon, Cordes, AZ 86321
```

Figure 2.
Search designs.

this dBASE II shines. You can design just about any selection criteria you can imagine. You have all the normal comparisons: equal to, not equal to, greater than, less than, greater than or equal to, and less than or equal to. dBASE II also adds the logical tests, NOT, AND, and OR. You can search for substrings—characters inside a word or phrase. Three examples of simple search designs appear in Figure 2.

In a complete accounting system, you might want to create a printout (report) showing all customers who live in the 22018 ZIP area, owe you more than $125, and purchased nothing in the last month. REPORT does this compilation and displays it. Add PRINT and you will have hard copy. The Report command automatically takes you through a series of questions for ordering up headings, subtotals, and so on. This "form file" can be reused, too.

You can SORT files (say, alphabetically). dBASE II also has an INDEX function, which sorts files on a key word or words, to make finding an individual data item much quicker. Up to seven index files can be maintained for the database in use, at any one time, and an index file in use is automatically updated when records are added.

The Find command goes to the first record that fits your selection conditions, regardless of where the pointer is set in the database. Once it finds a record, dBASE simply reports: RECORD 11822. To view the record, you have to type DISPLAY. It seems that once you ask to find something, you should see it immediately—a minor annoyance.

dBASE II allows you to position items in printouts or on the screen through the use of SAY@. You need only supply the coordinates. Current versions of dBASE II come with a program called ZIP that does this automatically, making report printouts and input screens much easier to design.

dBASE II lets you JOIN two databases, merging their fields into one file. However, they warn against creating a new database larger than the 65,535-record maximum.

Weaknesses

For the scientist, dBASE II may not be appropriate, as some standard calculator-type functions are not included. There is no graphics capability within the program, although accompanying literature mentions an add-on program for business graphics from Software Plus.

If you APPEND data from one file to another, newer file, this word of caution appears in the manual: "If the new records are smaller than the old records in the Use file, then the new record is padded on the right side with blanks; if the new records are longer than the Use file records, then the newly appended records are truncated."

The Manual

The manual has over 350 8½-by-11 pages, including a 12-page index, with boldface page numbers to indicate the best entry point for a particular topic. The book is logically divided into two parts. The first is a readable and easy-to-follow description of a sample program written by a user of dBASE II, along with a complete accounting system in the language. It's comforting to know that other people can do so much with a program that may seem intimidatingly different. The second half of the manual is by the author of dBASE II, Wayne Ratliff, and offers excellent explanations of the expressions and commands of the system.

Getting Started

If you're considering dBASE II, Ashton-Tate will sell you the program, sending along a sample disk for you to try. It only allows 15 records per file, but you can do everything "in miniature" with this disk, on a trial basis. If you like it (you have 30 days to decide), you can unpack the "real" disk and keep dBASE II. If not, you can send it back. Good idea.

Should you have trouble learning or using dBASE II, or if you're looking for already-written applications programs, everything from books to seminars to user groups are appearing on the scene. Watch the ads.

The Bottom Line

For the person who wants to buy a program and have it do one specific task, dBASE II is a waste of money.

For the person who does not want to learn what amounts to a new programming language, dBASE II is a bad buy.

For the person who wants to slip in a disk and have a program up and running in ten minutes, forget dBASE II.

But for those (probably in business environments) who want to create a complete information system, designing in the specifications and limits, with the knowledge that things can be quickly changed later on, dBASE II is ideal. For the programmer who needs to do custom applications based on a standard format—but one that pro-

vides almost unlimited flexibility—dBASE II is terrific. The person willing to spend the time to learn dBASE II will soar to new heights in microcomputer application.

In our own small business, I wrote the accounts receivable system, the payroll system, the inventory system, and so on—all in Basic. They all work, but the thought of integrating them into one huge database is almost inconceivable to me. I also suspect that any Basic program that would attempt such a thing would very quickly run through my 64K of memory.

dBASE II, on the other hand, is designed for just such an application. It's flexibility alone—to do what you want it to—is worth the price.

Don't expect it to be easy, especially the Command files, the ones that will be actual "programs." And don't expect to get things perfect the first time.

Consider the complexity as an opportunity. Do expect to be able to change things later on as you need. Do expect the system to perform as advertised. And do expect to do things—soon—you never dreamt of yesterday. Expect, frankly, to be amazed! ■

A note of thanks to Pete and Gary of the ComputerRoom in Prescott, AZ. They were kind enough to allow us to use one of their Apples running CP/M to work with dBASE II.

Editor's note—

Ashton-Tate constantly upgrades dBASE II. The current version of the program (2.3) has been on the market for about one year. The upcoming version (2.4) will be available about the same time this issue of inCider *is published.*

Version 2.4 will contain some important improvements. For instance, it will allow you to clear just some of the memory variables you have in use, rather than having to release them all.

Another welcome addition is the availability of a Help function inside dBASE II. You can even incorporate your own comments into the Help file when you write your Command files. This gives the dBASE II programmer more flexibility in communicating with the end-user—to lend a HELPing hand.

Yeah, But Why'd You Name It Beagle Bros?

It's the New American Dream—buy yourself a computer, quit your job and start your own software company.

by Bert Kersey

Well, you've got to name a software company *something*, and "*DATA*-Everything" and "*SOFT*-Everything-Else" had been taken, even way back in 1980. My wife Sharon and I were driving home from San Felipe trying to concoct a name "that people would remember" when I recalled Walt Disney's stripe-suited "Beagle Boys," who spent the better part of each day escaping from prison and tunneling into Scrooge McDuck's vault. I changed "Boys" to "Bros," and launched a little business.

My computer background was zero in the Spring of '79 when I brought home my first machine. I had given up on the three big reasons I was told I needed such a toy: balancing the checkbook, storing recipes, and controlling the house lights. I decided that being fascinated by typing on a TV screen was reason enough. I brought home a TRS-80 Model I for $600 (lowered to $500 the next day, by the way) and was captivated. Unfortunately (at the time), it broke down three times in two weeks, so I got my $600 back.

But I was hooked. I was about to buy my second choice, a PET, when the dealer talked me into an Apple "minus" with 16K, Integer Basic and cassette I/O (Apple disk drives and Applesoft were things of the future). I still owe him for that piece of advice. As he predicted, the PET and TRS-80 software of that era has since become obsolete, while even the latest Apples can handle my ancient (4-year-old!) programs.

In those days I was a successful, but bored, free-lance graphic designer, pumping out mail order ads, billboards, logos, brochures, and such, from my office at home. When I started playing with my Apple, I realized that advertising wouldn't be such a dreary occupation if I were advertising my own company.

I made Big Move #1 and retired (with no gold watch or anything), then devoted 14 hours a day to "working" on my Apple, with some kind of income-producing business as the goal. I was rarely seen outside the house. Sharon supported us during that unsure year—our third "switch" where one of us supported the other during a career change. Sharon was often less than thrilled upon coming home each day to find that I had blown the day teaching my Apple to write her name backwards, or draw a blue lo-res cow.

For a beginning programmer, writing games was the perfect learning vehicle. I wrote games; and then I expanded them into bigger games. In a year of 14-hour days, you can write a lot of games, so when it came time to make Big Move #2, placing my first ad, that's what I had to offer the world. (Some of them are pretty good, if I do say so, but to this day, I can't *give* away a copy of "Tic Tac Fooey.")

My advertising dilemma was this—how do you sell computer games, good *or* bad, by mail amidst the sea of advertising already in the magazines—especially if your name is "Beagle Bros?" Well, chapter 1 in my *How To Get Rich In Mail Order* book says, "Offer them something free," (even if it isn't free). I wrote a 36-page Apple Tip Book and printed up a chart of every Basic and DOS command I could find. My ad headline, instead of saying "Buy Our Wonderful Games," said "FREE APPLE TIP BOOK AND COMMAND CHART" (and, in small type, "With Game Purchase"). My objective was to offer people something they needed, and could picture having.

It worked.

The ad was reworked and reworded to death, and finally placed—a quarter-pager in *Creative Computing*. Chapter 2 in *HTGRIMO* says "Every ad must have a coupon," and mine had four boxes to check: Applesoft, Integer, Disk and Cassette. I had naively written all 16 of my games in Integer Basic, because I figured that was the language everyone had (re-

Bert Kersey is the Beagle brother. When he's not selling software, he's enjoying his Jaguar. Send correspondence to him at 4315 Sierra Vista, San Diego, CA 92103.



member, I never left the house). And *surely* not too many people had blown $600 each for those new-fangled disk drives, so I stocked up on blank cassettes.

I had finally "thrown my hat over the fence" and couldn't wait to see what was going to happen. Two agonizing months elapsed between placing the ad and seeing it in print, but by golly, one day I got an ORDER! And then another . . . then two and three a day.

I was on my way to "Getting Rich In Mail Order." The big surprise was that 80 percent of the orders were for disks and 80 percent for Applesoft, with the categories overlapping. I became *very* busy converting Integer to floating point, making copies of disks and cassettes, sticking on typed labels and driving to the post office.

Not much time passed before people began returning disks, saying they wanted "16-sector" instead. (Of course, they kept my free book and chart.) But wait! I had never heard of 16-sector. What was happening out there? Before long, each program I sold had to be stocked in both 3.2 and 3.3 Applesoft Disk, 3.2 and 3.3 Integer Disk, and Applesoft and Integer Cassette (one guy even ordered a 3.3 Cassette!). And all of my friends were urging me to convert my games for other computers! Things are simpler now with 3.2, Integer, cassettes and thoughts of going in other directions than Apple all being relics of the past.

Up until this time, I didn't quite know what a utility disk was, but a lot of people had written asking if Beagle Bros had any. One day I read that you could reword Apple's DOS commands by poking characters into memory. That's when I decided to try my hand at writing one of those "utilities." Soon after that we had, by our standards, a hit on our hands, and Beagle Bros was (and still is) in the Apple utility business. In a few months we had created DOS Boss. My new friend Jack and I spent weeks blindly manipulating Apple's DOS, mostly cancelling machine-language instructions with NOPs to see what kind of crash would result.

Any successful business results from putting together lots of pieces, and it still amazes me *how many* pieces there are. I highly recommend that you stick with what you're good at, and hire someone else to do the rest. If you're a programmer, program. If you're a marketer, market. I evolved into both professions over a twenty-year period, so I'm cursed with doing it all—with the help of Sharon and our highly-trained crew of bag stuffers. The last time I checked, Beagle Bros Inc. was averaging over 200 disks sold per day, 90 percent to stores and 10 percent mail order. We still make it all happen at home, and we're doing our darn'dest to keep it that way.

We get about one piece of unsolicited software for review a day, ranging from Hurricane Tracking programs to Apple Diet Plans. Almost without exception, these disks are "not quite finished" and the documentation, if any, is in "rough draft" form. I heartily suggest that you debug your disks before you send them for review. Also, send an inquiry letter ahead to see if the software publisher is even interested in what you have to offer.

And, if you're going to start your own software company, remember that the names "DATA-Everything" and "SOFT-Everything-Else" are taken. . . . And so is "Beagle Bros." ■

If you don't find our products at your Apple Dealer, tell him to phone Beagle Bros, 714-296-6400, **OR** his favorite software distributor.

NEW!

Flex Text
70-Column Text Utility
by Mark Simonsen

Flex Text is a unique utility that lets you print variable-width text on Apple's hi-res screens in normal 40-column format, 20-column expanded or 56- and **70-column condensed** characters. Character widths may be mixed as you like for emphasis. Flex Text understands **normal Applesoft** Basic commands, including Home, Inverse, Normal, Vtab 1-24 and **Htab 1 through 70**! It also supports text window pokes and scrolling, so you can program normally, but with the **ability to add text to graphics**, or graphics to text! You can even run your existing programs using these features!

FLEX TEXT IS COMPATIBLE WITH DOS TOOL KIT© FONTS

Enter up to NINE font names in Flex Text's boot-up program for easy ctrl-command access. **Upper & lower case** in any character-width without hardware. All characters redefinable with a **text character editor**. Toggle between "normal" text screen and both hi-res pages. Compatible with Neil Konzen's Program Line Editor© and GPLE.©

$29.50 Machine language. Unprotected. 48K min. Peek/Poke chart included. Condensed character display requires a monitor (instead of a tv) for best results.




```
10 REM HI-RES NUMBER GENERATOR
20 SIZE=5: SCALE=SIZE: REM NUMBER-HEIGHT
30 HGR: HOME: POKE 232, 0: POKE 233, 3: ROT=0
40 FOR A=768 TO 830: READ B: POKE A, B: NEXT A
50 N=N+1: N$=STR$(N): X=99: Y=0
60 FOR A=1 TO LEN(N$): HCOLOR=0: DRAW 8 AT
   X, Y: HCOLOR=3: DRAW VAL(MID$(N$, A, 1))
   AT X, Y: X=X+SIZE+SIZE: NEXT A: GOTO 50
70 DATA 20, 0, 24, 0, 27, 0, 31, 0, 35, 0, 39, 0, 44, 0,
   49, 0, 52, 0, 57, 0, 53, 62, 36, 0, 49, 38, 0, 53,
   55, 61, 0, 53, 23, 37, 0
80 DATA 46, 38, 52, 0, 61, 46, 62, 5, 0, 61, 54, 37, 7,
   0, 53, 38, 0, 54, 37, 60, 46, 0, 53, 39, 53, 62, 5, 0
```

DOS Boss
Disk Command Editor
by Bert Kersey & Jack Cassidy

A classic Apple utility you will ENJOY! Rename DOS commands ("Catalog" can be "Cat", etc.). PROTECT PROGRAMS; any unauthorized save- attempt produces a "Not Copyable" message. Also **List-prevention** and 1-key program-run from catalog. Custom catalogs: Change Disk Volume message to your title; Omit or alter file codes. Rewrite error messages: "Syntax Error" can be renamed "Oops!!" or anything you want! **Two books included**— Fascinating documentation and hours of good Apple reading!

Dos Boss's change features may be appended to your programs so that anyone using your disks (booted or not) formats DOS as YOU designed it.

$24.00
☐ Unprotected disk (32K/48K)
☐ The Dos Boss Book
☐ Beagle Bros Apple Tip Book #2
☐ Peeks & Pokes Chart



NEW!

ProntoDOS
High-Speed DOS Utility
by Tom Weishaar

ProntoDos is FAST, saving you time where it counts the most. This comparison with normal Apple-DOS speaks for itself—

FUNCTION	PRONTO	NORMAL
BLOAD HI-RES IMAGE	3-sec.	10-sec.
BSAVE HI-RES IMAGE	6-sec.	12-sec.
LOAD 60-SECTOR PROGRAM	4-sec.	16-sec.
SAVE 60-SECTOR PROGRAM	9-sec.	24-sec.
BLOAD INT/LANGUAGE CARD	4-sec.	13-sec.
TEXT FILES	(no change)	

MORE DISK SPACE: Booting ProntoDos frees up 15 extra sectors of Disk Space, almost a full track. To speed up your Apple, just boot ProntoDos or any disk you have updated with ProntoDos, and you're in business. You can even create new ProntoDos disks with Apple's normal INIT command. ProntoDos is compatible with ALL commands and performs normally (but FAST) with almost ALL programs.

$29.50 Machine Language. Unprotected. Peeks & Pokes chart included. All normal 3.3 disks are updatable

Tip Disk#1
100 Tip Book Tips on Disk
by Bert Kersey

100 programs from Beagle Bros' Tip Books 1, 2, 3 and 4— Fascinating tricks to make your Apple do things it's never done before! All 100 programs are listable, copyable and changeable; and each teaches another fascinating Apple programming technique. **Two different charts are included.**

$20.00
☐ Unprotected (32K/48K)
☐ Peeks & Pokes Chart
☐ Apple II Command Chart

BEAGLE BROS DISKS ARE UNLOCKED AND UNPROTECTED. THIS MEANS EVERY PROGRAM IS INSPECTABLE, CUSTOMIZABLE, IF YOU WANT, AND COPYABLE, GIVING YOU THE MOST FOR YOUR SOFTWARE DOLLARS. DON'T SETTLE FOR LESS.



Beagle Bros™ MICRO SOFTWARE

4315 Sierra Vista, San Diego, Ca 92103
714-296-6400

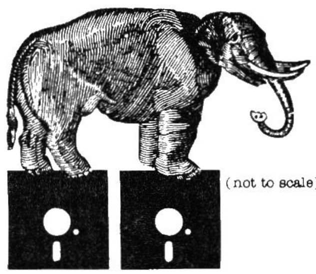
"APPLE" is a registered trade mark of You-Know-Who.

Utility City
21 Utilities on One Disk
by Bert Kersey

A best-seller since it hit the market, and a MUST for your program-development library. Take a look at the features— **List Formatter** makes properly-spaced & indented listings with page breaks; each statement on new line, if-thens and loops called out; a great de-bugger! **Multi-Column Catalog** in any page-width to any printer or CRT. Auto-post Run-Number and last-used Date in programs. Put INVISIBLE working commands in your listings. Access program lines in memory for repair & illegal alteration. Alphabetize & store info on disk. Run any program while another stays intact. Renumber to 65535. Save inverse, trick and **invisible file names**. Convert dec to hex & binary, or Integer to FP. Append programs. Dump text screen to printer...

21 LISTABLE UTILITIES TOTAL!

$29.50
☐ Unprotected disk (48K min.)
☐ Beagle Bros Apple Tip Book #3
☐ Peeks & Pokes Chart



(not to scale)

GOTO Your Apple Dealer.

Most dealers carry our software. If yours doesn't, he can have our disks in his store for you **within 2-3 days** by calling us or his favorite software distributor.

Or Order by Phone:

24-Hour **Toll-Free** Order Desk:
Visa/MasterCard/COD* orders, call
Nationwide: **1-800-854-2003** ext.827
California: **1-800-522-1500** ext.827
Alaska/Hawaii: **1-800-854-2622** ext.827
(ORDERS ONLY, PLEASE) *COD, add $3.00.

OR ORDER BY MAIL—

RUSH! The disk packages checked below plus the free 11x17 Peeks & Pokes Chart:

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ A.Mechanic	$29.50	☐ ProntoDos	$29.50
☐ Beagle Bag	$29.50	☐ Tip Disk#1	$20.00
☐ DOS Boss	$24.00	☐ Typefaces	$20.00
☐ Flex Text	$29.50	☐ Utility City	$29.50

(Add $1.50 Shipping, any size order. California, add 6% tax.)

NAME$= ______
ADDR$= ______
CITY$= ______
ZIP$ ______

VISA / MASTERCARD Include Card Number & Expiration Date.

Mail U.S.check, money order or Visa/MasterCard numbers to:

BEAGLE BROS, DEPT.I
4315 SIERRA VISTA
SAN DIEGO, CA 92103

All Orders Shipped Immediately.

Please add $4.00 for shipping outside North America. COD orders add $3.00. California residents, add 6%.

Circle 179 on Reader Service card.

A Step Forward: The List Handler

Fast, easily mastered and powerful, this list manager holds many pleasant surprises... including its price.

by Bill DeVille

The List Handler is a recent release from Silicon Valley Systems Inc., 1625 El Camino Real, Belmont, CA 94002, for Apple owners who need to manage large lists of information, from mailing lists to customer files, personnel records, library card files, etc. Users of the Word Handler word processing program should put this on their must-have list. Besides retrieving selected and sorted information, it generates customized form letters based on the content of a list file. The program can also exchange data with other programs that use the DIF format for data exchange.

Program Description

The program is easy to use. The manual, *The User's Guide to the List Handler*, provides a step-by-step tutorial for each feature of the program. Anyone totally unfamiliar with the operation of a computer can, after a few hours spent on the tutorial exercises, become proficient in designing lists, data entry and editing, modification of an existing list, and manipulation of the information to provide form letters, mailing labels, simple columnar reports, and queries of the list information.

Helpful menus, displayed at the bottom of each screen, identify the command options available at each step, and provide cues to their use. Even the powerful field selector, logic operator selector, and sort order selector operations are defined on screen in understandable language. These operations use the arrow keys to page rapidly through the available options; when the desired option appears on the screen, it is confirmed simply by pressing the return key. No previous knowledge of the logical operators is required, nor do you have to learn special logic symbols to use the program. In this sense, the List Handler approaches the ideal of a "natural language" program.

Once the tutorial has been mastered, you should rarely, if ever, need to refer to the manual. The program guides itself to a large degree.

Upper- and lowercase text display is offered, even on an unmodified computer, with a screen display of about 73 columns. Alphabetic characters are displayed with true proportional spacing—for instance, the letter m is about three times as wide as is the letter i. The program accepts a simple, user-installable shift key modification to allow typewriter-like use of the shift key for capitalization.

Printout Information

The program's selection criterion for a printout uses the list information in any selected record field. It gives a choice of eight logic operations—"equal to," "equal to or before," "equal to or after," "equal to or between," "not equal to," "not equal to and before," "not equal to and after," and "not equal to and between." They are selected by using the arrow keys and may be in alphabetical, numerical or chronological order.

The information sort order is based upon the alphabetical, numerical or chronological data content of any field in the record, as well as ascending or descending order.

The printout specifies information from any or all fields contained in the records selected, and is not limited to the fields used for the selection criterion and the sort order.

Printing Form Letters

I use the program to generate customized form letters based on the contents of the list disk. It works well, especially using the Document Input from a form letter document created by Word Handler (it will also work with text files created by other selected word processing programs). When the form letter document is created, the variables to be filled out are entered as specified field names contained on the list disk. As a form letter is printed out, these variables are filled by the infor-

Address correspondence to Bill DeVille, Route 6, Box 178, Baton Rouge, LA 70815.

Tired of Waiting for Your Disk-Drive?



THE SYNETIX SOLID STATE DISK EMULATOR FOR THE APPLE, INCREASES SPEED UP TO 1000%.

The Synetix Solid State Disk emulator operates identically to the standard disk drive, but **responds much faster** for programs using disk I/O. The SSD also saves wear on your mechanical drive and diskettes.

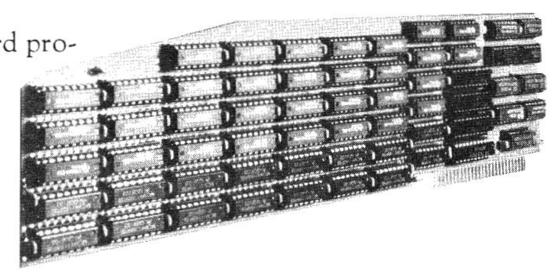
The Solid State Disk Emulator is **compatible with** DOS 3.3, Pascal, SofTech p-Systems IV.O, IV.1, and CP/M operating systems. Call us for a complete list of compatible software and benchmark results on aforementioned systems.

Database management, word processing and file manipulation programs can run up to 1000% faster.

A single board fits any slot and can respond as one or two mechanical disk drives. The SSD requires no external modification and uses only 150 ma power supply or per board.

Two Models are now available at new low prices. The Model **2201 (147K)** single disk is now only **$395.00.** The **Model 2202 (294K)** dual Disk is **$695.00.**

The Model 2201 (147K) **is upgradeable** to a Model 2202 (294K) by you or by Synetix.



Contact your local dealer, or call:

Synetix Micro Products Inc.
15120 N.E. 95th Street
Redmond, WA 98052
1-800-426-7412

Apple and Apple DOS are trademarks of Apple Computer Inc. CP/M is a trademark of Digital Research.

Circle 10 on Reader Service card

mation contained in the specified fields of each record selected. Any or all of the field names contained on the list disk may be included in the form letter, and each may be repeated.

The full powers of the selection criterion and the sort order are available in the fill mode, and the printing routine can adapt to any page size desired to print the document. One available option is to have the routine pause after typing each form letter (to allow insertion of letterhead stationery into the printer), or to print continuously on fan-fold paper. This feature is not limited to printing form letters but can be used for reports that require the "filling in" of variables from information contained on a list.

Computer File Drawers

List Handler is a flexible, fast and powerful system for creating, storing and providing selected and sorted retrieval of information. This information (alphabetical characters [text], numbers or dates in any combination) is organized into *records*.

A record consists of *fields* where the actual information content of a record is entered. These fields are created by the user—Last Name, Subscription Expiration Date, Dewey Decimal Number, Account Number or any information to be stored and retrieved.

List Handler allows exceptionally large record sizes compared to other programs. A record can be a maximum of 4000 characters in length, with up to 255 fields within a record. In turn, each field can have 256 characters (less the length of the field name, and a bit of space required to index the proportionally-spaced characters in the display).

A list disk created by List Handler may be thought of as a file drawer, which contains file folders (records). Each of these file folders holds a series of forms containing information fields). Just as a busy office fills a file drawer with folders, and expands into additional file drawers, List Handler allows expansion of file storage onto additional disks. Since a List Handler list disk can hold up to 3000 records such as name and address records (fewer if a record has a great many fields, each of which contains several lines of text), this program provides an effective "file drawer" space that is up to several times larger than many other programs available for the Apple.

Large List Use

It is a simple matter to develop a large list containing thousands of records. The program package provides a utility program descriptively named "Extend Old List Onto New Disk." This utility initializes a new list disk containing all the field names of an existing list. If an extension disk has been previously prepared, it can be inserted during List Handler's "Add Records" mode whenever the current list disk gets filled up. When one "file drawer" is crammed full, it is easy to start filling up a second one, and so on.

If you have developed an extended list that requires two list disks, and have two disk drives, both can be logged on at the same time. This is one of List Handler's strong points as many Apple programs do not allow multiple disk access for searches and sorts. The program allows logging on up to a maximum of eight disk drives on-line simultaneously. Since up to 3000 records can be stored on a single side of an Apple floppy disk (depending on the number of fields and the amount of information contained in the fields used), the program provides for on-line access of up to 24,000 records, using Apple floppies!

Of course, as a list grows and extends onto multiple drives, the time required to sort a printout of information from the list increases. If, for example, two drives are logged on to access a two-disk list, and a sorted printout is requested, the program will search through both disks sequentially to check all the information requested, and arrange it into the desired sort order. But anyone who is familiar with the operation of other file management programs will be amazed by the speed at which List Handler can perform a multi-disk sort of a list whose records are in random order.

A Mailing List Application

My largest List Handler application is printing mailing labels from a list containing 2800 records. This complex mailing list is actually 13 mailing lists

"I decided to transfer the 2800 record mailing list over to the Word Handler... without any retyping of information..."

in one, since it contains that many individual selection criteria for doing mailouts.

My mailing list was originally created using the DB Master program. I have a lot of respect for DB Master, and I continue to use it for purposes that List Handler cannot do (yet), such as applications that require mathematical functions—e.g., subtotals, totals, computed fields and statistical analysis of data. But a mailing list program requires none of these features. All I need for a mailing label printout is a selected and sorted (usually Zip Code) operation on the mailing list.

When I first got my List Handler program package, I experimented with the sample mailing list of about 400 records contained on the "Tutorial List" disk. This list contains a number of fields (name, Zip Code, dates, phone numbers and so forth). I could set up a sort on any field in the record. Even though the field contents were randomly entered, it took the program only 60 seconds (give or take 5 seconds based on other variables such as search conditions and fields selected), before it started printing the requested mailing label format. Once it started, it drove the printer rapidly to put out a batch of about 65 labels (this number may vary a bit, depending on the number of lines in your label). The program then goes back to the disk for a few seconds to pull up the next batch, and so on.

Because I had a number of DB Master files of varying sizes, I compared the two programs for output (a mailing label printout with the same number of items and lines) and for numbers of records (up to a few hundred records). The best case for List Handler was slightly more than *five times* faster than DB Master! The worst case was three times faster. To make the point even more apparent: DB Master was accessing the files in key sort order (the records had been stored on disk in the same order in which they were being retrieved). List Handler was doing its sorts from a standing start (working with records stored in random order on the disk). If I had requested a sort from DB Master other than one based on the primary keys of the records, I would have had to go through another step, actually recopying the file onto new disks before requesting the printout. The comparison is even more favorable to List Handler than might be apparent at first.

Mailing List Transfer

Following these experiments, I decided to transfer the 2800 record mailing list over to List Handler. This was done without any retyping of information contained in the DB Master file, except for the field names. Using the DIF utility provided by DB Master, I translated the mailing list file into a number of blocks of records (typically, about 300 records each, based on a Zip Code range selection of records). Then I created a new list disk using the List Handler Create New List utility, making certain that the field names in the new list were *exactly* the same as the field names in the DB Master file.

Using List Handler's utility program for translating DIF files, I transferred the DIF files that had been created from the original mailing list onto the List Handler list disk. This is where two relatively minor bugs appeared in the List Handler DIF utility program.

First, when DB Master (which uses fields of defined length) created the intermediate DIF files, it inserted blank spaces into the DIF text file where the original fields had not been filled to the maximum defined length. When the List Handler DIF utility captured these DIF text files, it slavishly inserted all these blank spaces (about 100 extra characters per record) onto the new list disk. I don't know if these blank records were created by the DB Master utility, or by the List Handler utility.

Transfer Results

I was disappointed in one respect. The original mailing list took roughly two and a half disks for the 2800 records, in the DB Master format. The transferred mailing list took roughly the same amount of space, so that it spilled over onto three list disks. My calculations indicate that the new list could be contained easily in less than two full list disks—and, very possibly, on only one list disk—if I knew how to get rid of all those unnecessary blanks stored on the list disks.

Was it worth it? Yes, for two rea-

sons. First, the time for printing mailing labels has dramatically dropped. With DB Master, printing out the entire list (which is required at least once a month) took more than six hours. This was complicated additionally by the fact that only one data disk at a time could be on line, so that someone had to watch the process to change disks occasionally. With List Handler, the entire printing job takes slightly less than three hours. The improvement is even more dramatic for the shorter mailing lists (based on subsets of the entire list). For the shortest list, which contains only a few names, the speed increase is about fivefold. List Handler works quickly because it can grab all the hits in one pass through the file, store the data in memory, and then print them in rapid sequence.

Although the list was originally in Zip Code order, it has already become largely randomized because several hundred entry changes and edits have been made to the mailing list. As the file structure has changed from ordered to randomized, no significant slowdown has been observed.

"Another major advantage is that our clerical staff is much less terrified of List Handler than of DB Master."

There are other advantages to List Handler. Entries or edits are in upper- and lowercase, and look better—even for mailing labels. List Handler recognizes a simple clip-on shift key modification for upper/lowercase typing. Another major advantage is that our clerical staff is much less terrified of List Handler than of DB Master. It is a much "friendlier" program. Finally, all the list information is now readily accessible for form letter generation. It was well worth the changeover for us.

Creating a List

List Handler requires a certain amount of organization as any good office filing system does. This organizational structure is easy to design and to change in mid-course. You define the structure when creating a new list (or modifying an existing list) by selection of the field names contained in a record.

The field names should be so descriptive that other users of the information system will understand it, and know the proper fields new data is entered, or existing information is found.

List Handler does not require you to specify field lengths or types when a

"List Handler allows you to manipulate information in ways that are cheaper, and saves a great deal of time..."

list is created. Yet it is capable of logical functions that examine numerical or date fields, as well as alphabetical fields. Some conventions are expected. Dates can be entered as MM/DD/YY, (or several other such formats) as long as there is a spacing such as '-' or '/' between the month, day, and year fields.

This is a very user-friendly approach. Although it does not provide some of the error trapping capabilities of other programs, which would not accept entry of an alphabetical character in a numerical field, I am not unhappy. Most programs provide only minimal error trapping in designing a new database at the expense of a good deal of added complexity. My experience shows that a good deal of care in entering data into a record cannot be substituted by any other approach. List Handler expects you to enter the data correctly—just as you would when using a typewriter, or a pen. The added advantage is that almost anyone can create a new list, without trying to predict field lengths, or knowing anything at all about field types. I know that a child can do it, because my nine-year-old daughter, Nique, easily designed a list using List Handler.

Only one list can be contained on a List Handler list disk. This is like the practice many business offices use, of reserving one file drawer for personnel records, another for client records, and so forth.

File Management Purposes

The purposes of filing information are twofold: to provide secure storage of the information until it is needed, and to access the desired information when it is needed. List Handler serves both purposes admirably. It provides easy and reliable data entry and disk storage functions, and an easy routine for making backup copies of list disks for improved storage security.

This program really shines in locating information. It rapidly scans the contents of a number of file cabinets (to continue the analogy); pulls information out of the file folders based on a selection criterion, with specified logical conditions; and then sorts it into the desired order, based on alphabetical, numerical or chronological order.

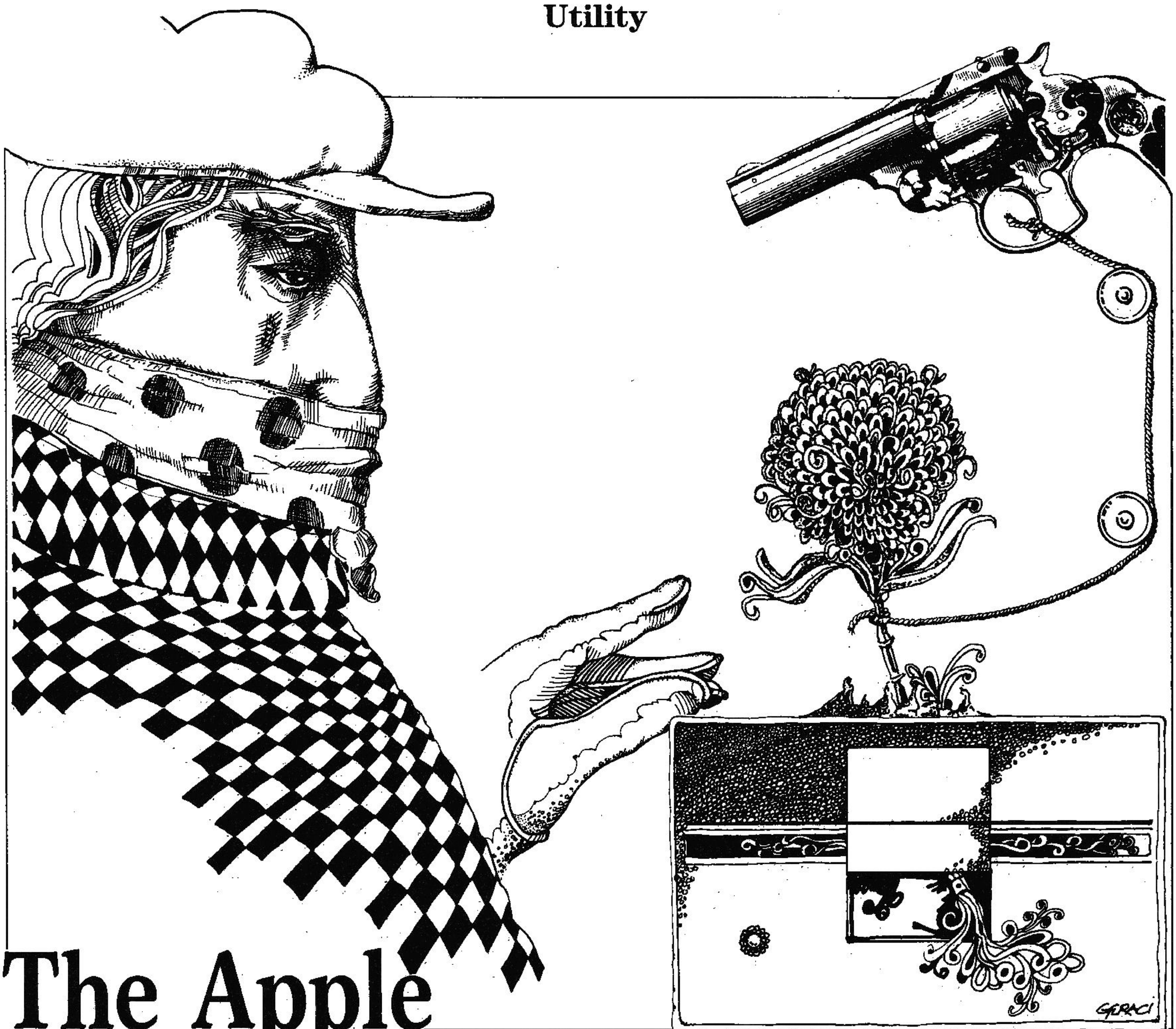
List Handler Advantages

List Handler can make an Apple genuinely useful for managing large information files for business, science, technology and other uses. The program has a great deal of power to select and sort information from a list, and to allow use of eight logical operators based on the alphabetical, numerical or chronological (date) contents of a selected field.

In short, List Handler allows you to manipulate information in ways that are cheaper, and saves a great deal of time compared to normal clerical operations. This inexpensive program allows useful operations on large amounts of information. Many things that can be done easily with this program would have been literally impossible in a small business or professional office, until very recently (other than by a service contract with an outside computer service company or time-sharing system).

List Handler does not offer features that a more sophisticated database management and reporter package contains, such as the ability to do computations, use computed fields, and print out highly structured reports with headers, subtotals, totals, etc. (Note, however, that it can turn out useful columnar reports, even without these features.) I predict, however, that Silicon Valley Systems will continue to add features to what is already a very powerful program. Watch this one!

List Handler competes in the marketplace with a number of good file-handling programs available for the Apple, such as PFS, VisiDex, and Data-Fax; and with non-computational uses made of database management programs such as DB Master or The Data Factory. It has advantages over other programs I have seen. For many of the applications it is now capable of, such as file management, mailing label printing, card files, and automatic form letter generation, List Handler is probably the best bargain on the market at $89.95, especially in view of the many features provided, including form letter capability, and the unusually large and flexible record size and field size. ■



The Apple Combination Lock

by Harry White

Omigod! Look at that!

In response to Joe's distress I turned from my window gazing to find him pointing at his Apple's CRT. There, nicely centered on the screen, was a graphic suggestion that he do something to himself which was biologically impossible.

"Where did that come from?" I inquired.

"It came from that attendance file program—or what's *left* of it. The students have been at it again," he said with a shake of his head.

I had stopped by the public school to pick up Joe, my teacher friend, late Friday afternoon for a camping trip. He had a little work to finish up and I was loafing around his sixth grade classroom when the preceding incident took place.

Before we left that evening we discovered that his attendance files were intact. The student had only added to them, leaving the obscene message for Joe. Little damage had been done, but it stimulated an ongoing conversation about such problems and some possible solutions.

At the end of the weekend camp-out we had developed a philosophical approach and some practical solutions.

Our philosophy shared its base with the law enforcement observation: You can expect only to slow down a burglar, not defeat him entirely. Also, the strength of a burglar's motivation and the complexity of his ingenuity is directly proportional to the worth of his goal. Guarding a cherry pie set out to cool on a window shelf and protecting Fort Knox elicit somewhat different security measures. The student's unpleasant act hovered just above pie snatching but well below a Fort Knox heist.

Some Software Solutions

Joe needed a system that was easy to implement but would defeat casual mischief. The following security method performs this task and has several variations. It is a software implementation of some hardware copy protec-

Address correspondence to Harry White, 7495 West 81st Ave., Arvada, CO 80003.

```
100 TEXT:HOME

120 VTAB 10: HTAB 15: PRINT PDL(0)

130 VTAB 10: HTAB 23: PRINT PDL(1)

140 VTAB 10: HTAB 15: PRINT "    "

150 VTAB 10: HTAB 23: PRINT "    "

155 IF PEEK (-16287) < 128 GOTO 120

160 IF PDL(0) > 160 OR PDL(0) < 150 OR PDL(1) > 130 OR
PDL(1) < 120 THEN GOTO 500

170 HOME: PRINT "IT WORKS": END

500 HOME: END
```

Program listing. Software security program.

tion devices that plug into the Apple's gameport. You need only a working knowledge of Applesoft and DOS commands to make it work.

Key in the short program in the listing to see the concept in its simplest form. Lines 120 through 150 display the settings of the paddle controls near the center of the screen. Paddle 0 is on the left, paddle 1 on the right. Line 155 checks to see if the button on paddle 0 has been pressed. If not, the program is sent back to line 120. At this stage all you see are the settings on the screen, which change as you twist the knobs.

Now press the 0 button. If knob 0 is not set between 150–160 and knob 1 is not set between 120–130, then the screen clears. You are left with a blinking cursor and the program has ended as directed by lines 160 and 500 after testing the paddle readings. This occurs if either one or both of the knob readings are out of the prescribed ranges.

If you wish to discourage the culprit rather than end the program, line 500 might look like this:

```
500 HOME: PRINT CHR$(7):GOTO 500
```

You have created a burglar alarm. If the paddle settings are incorrect, the Apple will beep until an operator intervenes. However, if you have set the knobs properly, execution continues to line 170, which prints the message, IT WORKS. The program would then proceed to its conclusion.

You have used the game controller settings to determine whether the program continues or ends—a combination lock on your work.

You can, of course, change the parameters in line 160 to whatever combination you wish. Note the range of settings, however. You might test for a single number rather than a range of 10, but the Apple paddle knobs really aren't that stable. You can probably cut the range to 5 but as you examine some of the expansions below you'll see the need for a little insurance.

A Stronger Lock

This simple device will defeat only the most inexperienced novice for but a short time. If the program ends he'll list the program and see what you've done, or he may have noticed the display of the paddle readings and have seen through your plan from the start.

There is a way to prevent the listing of a program which has been run. Add this line:

```
10 POKE 214, 128
```

When your program (or, at least, line 10) is run this Poke changes the operation of the Applesoft command parser so that any command, LIST for example, is interpreted as RUN. This requires more sophisticated knowledge of the Apple internals than the vandal may possess.

Now delete line 10. It will prevent some of the following plans from working. Save the program to disk under the name Lock Test, for later investigation.

Your program is still on the disk and is there to be loaded, listed, examined and changed by your adversary. You've managed to slow him down, maybe even catch him with the "alarm," but Lock Test is still right there in the disk catalog—a temptation and an opportunity.

An Evil Lock

It need not be. Save Lock Test again under the name, Lock Test.1. You now have two copies of the program on the same disk, but under different names. This will prevent having to rekey the thing after you try the next variation. Change line 500:

```
500 PRINT CHR$(13);CHR$(4);"DELETE
  LOCK TEST.1"
```

Run the program with the wrong paddle settings. The disk will whirr and quit. If you examine the catalog, you'll see that Lock Test.1 is not there. You have deleted it from the disk just as the thief would.

You can use a similar device to erase the program from memory. Save the program again as Lock Test.1. Then add line 510:

```
510 DEL 0,510
```

Run it with the wrong paddle parameters again. If you list it now, you'll find that it has evaporated from memory as well as from the disk. You are really causing your burglar a great deal of trouble and he may either give up or not have the knowledge to attempt a resurrection of Lock Test.1 via a disk utility. You, of course, have made a backup copy—haven't you? If not, you should, even if you are not using any of these protection schemes.

There may be even more evil schemes you can devise, but we've gone about as far as Applesoft will carry us. You disk jockeys can probably include the combination lock method in a modified DOS to prevent a disk from being booted or even read without the game knobs set properly.

Hidden Locks

However, you can provide a little misdirection in Applesoft. For example, you might want to scatter tests of the paddle knobs and the consequences throughout your program all in one long statement, not using GOTO 500. If you do this you run some chance of the knob settings being accidentally bumped and changed somewhere during the program's execution—then getting zapped as your work runs into one of the hidden lock statements. Storing the readings at a safe place in memory for later testing would avoid such a disaster:

```
157 POKE 6, PDL(0): POKE 7, PDL(1)
```



The memory locations 6 and 7 appear to be unused by Applesoft, DOS and the monitor except for the "Sweet 16" routines. Check out your Apple manuals. Such a line might read:

```
300 IF PEEK(6)□160 OR PEEK(6)•150 OR
    PEEK(7)□130 OR PEEK(7)•120 THEN
    HOME: END
```

. . . etc.

The fact that those memory locations, 6 and 7 (8 and 9 are also unused), are free permits the use of a "key disk" separate from your program disk as a further protection. Consider putting lines 100 through 157 on one disk. The paddle values would be poked into memory at 6 and 7. Your protected program on another disk, beginning with a line such as 300 or lines 500 and 510, would run to completion only if the key disk program had successfully preceded it. Those free memory locations will survive the run of your program and, in fact, will survive a PR#6 in DOS under most circumstances. Try it out.

By now your vandal may be so frustrated that he will decide to vent his ire and INIT your program disk. But that's OK; you have your backup—don't you!?

Joe tried variations of these methods over a period of several weeks and found that they worked to his satisfaction. His favorite system combined deleting the file from both memory and the disk and setting up the Apple's speaker to scream for attention by leaving a single line in memory containing an endlessly beeping loop.

"By now your vandal may be so frustrated that he will decide to vent his ire and INIT your program disk."

He had some words of advice and some more suggestions. You must certainly not lose track of the combinations and, to repeat, you must have made backups of any material affected by the locks. He emphasized that the best plan was to save the paddle settings to memory as early in the program as possible, rather than relying on the relatively unstable Apple game controls to remain where they were set. He elaborated on the lock to make it even more like a combination device by using a series of settings on one paddle, then testing for each parameter in turn.

A Hardware Solution

The winter passed and I didn't see Joe for a while. As spring came I stopped off again at the school to plan some future camping trips with him. He was busy with the Apple once more, but I noticed that the game paddles were not connected. They were lying over on the corner of his work table.

When he paused in his work I inquired if he had given up on the combination lock.

"Not at all. Worked fine. Just improved on it a little. Made it easier."

He turned the Apple off when he had finished and motioned me over to watch as he pulled the lid off the machine. He reached down to the game port socket and pulled, then proudly showed me something held between thumb and forefinger.

"See. I turned it into a hardware solution. Same principle, though."

What he held was a 16 pin IC socket with a tiny 10k ohm resistor added to it. One lead of the resistor was stuck in the hole over pin 1 and the other was inserted in the hole over pin 6. He explained that pin 1 was the power supply on the port and pin 6 was the other connection for paddle 0. By adding the resistor and plugging the socket into the empty port he had created a very stable setting on paddle knob 0.

On his Apple a test of knob 0 always returned a 20 when the socket was in place. By using a test of that paddle as described above, the lock worked just as it had via the software schemes we had worked out. He was planning to use a second resistor on paddle 1 when he got around to it. All it took for it to work was for him to plug the socket with the resistors attached into the empty game port. He was planning on making or purchasing a game port expander to make the task a little easier, perhaps adding a zero insertion pressure socket to save wear and tear on the parts.

Joe wasn't a hardware nut, but he knew how the gameport knob works. It converts analog input from the knob into digital information which the computer can use to return numbers from 0 through 255. The knob controls a 150k ohm potentiometer, a variable resistor, which sets a timer in the Apple. The timer counts down. The count is directly related to the amount of resistance introduced into the circuit by the potentiometer. Different knob settings, different resistance values, different counts, different numbers returned by PRINT PDL(0)—in that sequence. Joe's 10k ohm resistor emulated a controller knob set so that the timer routine produced a 20. A 15k ohm resistor would give a different result. See page 100 in your *Apple Reference Manual*.

Certainly the Apple combination lock doesn't copy protect your disks, but it will slow down all but the most determined vandal. It is sufficient protection in many situations and easy to implement. Such a method is worth adding to your Apple techniques. ■

Plan Your Retirement
With the Power of VisiCalc

You can't escape the inevitability of death or taxes. But maybe this will help you hold your own against inflation.

by G. R. Brieger

We are assailed daily by pronouncements from politicians and economists about the state of the American and world economy. By now the credibility of our economists is on a level roughly equal to that of our politicians. What they plan rarely works as it is supposed to; what they predict almost never happens. At best the economists are able to explain what has *already* happened.

Let's deal with the real world. Death and taxes have always been recognized as inescapable. We are faced today with a third inevitability—inflation. Thus, financial planning for retirement, though difficult and involved, is more important than ever before. The difference between proper planning and a lack thereof becomes the difference between a comfortable life of leisure and poverty.

This is where VisiCalc software (VisiCorp, 2895 Zanker Road, San Jose, CA 95134) comes in. It is ideal for solving "what if" problems, and the future is what we are concerned about.

VisiCalc is intended to be an electronic calculation sheet, but it really does a lot more. VC might properly be called a limited enclosed computer language. Unlike Basic and other computer languages, it operates in plain English and is easily understood by anyone with numbers facility.

The program consists of a matrix of 254 horizontal rows by 63 vertical columns. There are over 16,000 locations for values, formulas and labels. This saves time and avoids errors when set-

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
ROW NO:	YEAR	HUSB. SS	WIFE	PENSION	SPEND.	IRA	WITHDR.	LIQUID	TAXABLE	TAXES	ASS.BEG	% OF	YOUR	GRAPHIC DISPLAY	
		% INCREASE		/MONTH	INCOME		FR. IRA	ASS.	INCOME	/YEAR	OF YEAR	1983	AGE	OF ASSETS AT	
4					% INC.	% INC.		% INC.				DOLLARS		BEGINNING OF	
5		4.00	4.00		4.00	11.00		11.00						YEAR	
6															
7	1983			0	1600	100000		100000	11000	626	200000	100	60	#######	####
8	1984			0	1664	111000		91174	10029	496	202174	100	61	#######	####
9	1985			0	1731	123210		80739	8881	321	203949	100	62	#######	####
10	1986	600		0	1800	136763		68533	7539	211	205296	100	63	#######	####
11	1987	624		0	1872	151807		61463	6761	200	213270	100	64	#######	####
12	1988	649	243	0	1947	168506		53051	5836	100	221556	100	65	#######	####
13	1989	675	253	0	2025	182041	5000	46134	10075	396	228176	100	66	#######	#####
14	1990	702	263	0	2105	197066	5000	42655	9692	121	239721	100	67	#######	#####
15	1991	730	274	0	2190	208743	10000	38542	14240	864	247285	100	68	#######	#####
16	1992	759	285	0	2277	221705	10000	37686	14145	864	259391	100	69	#######	#####
17	1993	790	296	0	2368	215261	30831	36166	34810	4600	251427	100	70	#######	#####
18	1994	821	308	0	2463	208108	30831	50983	36440	5330	259091	100	71	#######	#####
19	1995	854	320	0	2562	200169	30831	66084	38101	6100	266253	100	72	#######	#####
20	1996	888	333	0	2664	191356	30831	81436	39789	6100	272792	100	73	#######	######
21	1997	924	346	0	2771	181574	30831	97810	41591	6900	279384	100	74	#######	######
22	1998	961	360	0	2882	170715	30831	114493	43426	6900	285208	100	75	#######	######
23	1999	999	375	0	2997	158662	30831	132291	45383	8600	290953	100	76	#######	######
24	2000	1039	390	0	3117	145284	30831	149597	47287	8600	294881	100	77	#######	######
25	2001	1081	405	0	3241	130434	30831	168028	49315	8600	298462	100	78	#######	######
26	2002	1124	421	0	3371	113950	30831	187676	51476	10300	301626	100	79	#######	######
27	2003	1169	438	0	3506	95653	30831	206943	53595	10300	302596	100	80	#######	######
28	2004	1215	456	0	3646	75343	30831	227453	55851	12300	302797	100	81	#######	######
29	2005	1264	474	0	3792	52800	30831	247308	58035	12300	300107	100	82	#######	######
30	2006	1315	493	0	3944	27776	30831	268399	60355	14300	296175	100	83	#######	######
31	2007	1367	513	0	4101	0	30831	288823	62602	14300	288823	100	84	#######	######
32	2008	1422	533	0	4265	0		310470	34152	4600	310470	100	85	#######	######
33	2009	1479	555	0	4436	0		312300	34353	4600	312300	100	86	#######	#######
34	2010	1538	577	0	4613	0		313222	34454	4600	313222	100	87	#######	#######
35	2011	1600	600	0	4798	0		313092	34440	4600	313092	100	88	#######	#######
36	2012	1663	624	0	4990	0		311749	34292	4600	311749	100	89	#######	#######
37							TOTAL TAXES FOR 30 YRS			162729					

Table 1. Sample retirement table.

This article appeared previously in the June 1981 issue of Microcomputing, *but has been completely revised and updated.*

Address correspondence to G. R. Brieger, 2716-174th NE, Redmond, WA 98052.

ting up tables. Any imaginable kind of arithmetic as well as transcendental functions can be performed within the confines of the 63-by-254 matrix.

Once you establish the format, VisiCalc fills the columns automatically. The calculations are done in seconds, right before your eyes, and the results can be output on your printer. Here, 132 column capability is desirable.

The accompanying sample tables show the information needed, including the constants used. VisiCalc recalculates the data instantly when any data, such as inflation rates or earnings on investments, is changed. The beginning dollar amounts in the columns can be varied to suit individual investments or pensions.

The Tables Explained

The variables tables (1-2) are for a couple, with the husband about to retire early, at age 60. He was born in 1923 and his wife is a few years younger. They have $100,000 in savings and investments, plus an IRA or Keogh account that can be rolled over. With the larger IRAs there is no pension (a lump sum instead); the smaller IRAs are supplemented by a lifetime pension.

This rollover account is tax-sheltered until the retiree reaches age 70½ (1993). At that time, according to present Internal Revenue Service rules, the fund must be liquidated over the expected lifetime of the couple. The money is fully taxable in the year it is withdrawn. Withdrawals can be made earlier, and possibly should be, to minimize income taxes.

Here is a detailed explanation of the tables, by column:

- Columns A and B represent the row numbers and years (1983 to 2012), respectively. The row numbers plus column identification in the top row make it easy to locate the data points. For instance, location F7 is the starting amount for spendable income (per month) after taxes. This amount is the most important figure for this application.
- Columns C and D contain Social Security payments. Changing constants results in recalculation of inflation/COLA (cost of living adjustment) increases or investment returns. The percentage increases used also show up automatically at the top of columns C, D, F, G and I.
- Column E holds the amount of a

	A	B	C	D	E	F	G	H	I	J	K	L	M	N O P
ROW NO:	YEAR	HUSB. SS % INCREASE	WIFE	PENSION /MONTH	SPEND. INCOME	IRA	WITHDR. FR. IRA	LIQUID ASS.	TAXABLE INCOME	TAXES /YEAR	ASS.BEG OF YEAR	% OF 1983	YOUR AGE	GRAPHIC DISPLAY OF ASSETS AT
4					% INC.	% INC.		% INC.				DOLLARS		BEGINNING OF
5		3.50	3.50		4.00	10.00		11.00						YEAR
6														
7	1983			600	1600	20000		100000	18200	1706	120000	100	60	####### ####
8	1984			600	1664	22000		97294	17902	1536	119294	100	61	####### ####
9	1985			600	1731	24200		93692	17506	1536	117892	100	62	####### ####
10	1986	600		600	1800	26620		88896	16979	1349	115516	100	63	####### ####
11	1987	621		600	1872	29282		90128	17114	1536	119410	100	64	####### ####
12	1988	643	241	600	1947	32210		90697	17177	1436	122907	100	65	####### #####
13	1989	665	249	600	2025	35431		93683	17505	1436	129114	100	66	####### #####
14	1990	689	258	600	2105	38974		96434	17808	1336	135408	100	67	####### #####
15	1991	713	267	600	2190	42872		99000	18090	1506	141872	100	68	####### #####
16	1992	738	277	600	2277	47159		101066	18317	1506	148225	100	69	####### ######
17	1993	763	286	600	2368	45675	6200	102719	24699	2600	148394	100	70	####### ######
18	1994	790	296	600	2463	44042	6200	108993	25389	2600	153035	100	71	####### ######
19	1995	818	307	600	2562	42246	6200	115262	26079	3058	157508	100	72	####### ######
20	1996	846	317	600	2664	40270	6200	121036	26714	3058	161306	100	73	####### ######
21	1997	876	328	600	2771	38097	6200	126687	27336	3058	164785	100	74	####### ######
22	1998	907	340	600	2882	35707	6200	132170	27939	3058	167877	100	75	####### ######
23	1999	938	352	600	2997	33077	6200	137433	28518	3518	170510	100	76	####### ######
24	2000	971	364	600	3117	30185	6200	141955	29015	3518	172140	100	77	####### #######
25	2001	1005	377	600	3241	27003	6200	146077	29469	3518	173080	100	78	####### #######
26	2002	1040	390	600	3371	23504	6200	149718	29869	3518	173221	100	79	####### #######
27	2003	1077	404	600	3506	19654	6200	152784	30206	4000	172438	100	80	####### #######
28	2004	1114	418	600	3646	15419	6200	154688	30416	4000	170107	100	81	####### ######
29	2005	1154	433	600	3792	10761	6200	155741	30532	4000	166502	100	82	####### ######
30	2006	1194	448	600	3944	5637	6200	155803	30538	4000	161439	100	83	####### ######
31	2007	1236	463	600	4101	0	6200	154718	30419	4000	154718	100	84	####### ######
32	2008	1279	480	600	4265	0		152310	23954	2226	152310	100	85	####### ######
33	2009	1324	496	600	4436	0		143956	23035	2226	143956	100	86	####### #####
34	2010	1370	514	600	4613	0		133374	21871	1846	133374	100	87	####### #####
35	2011	1418	532	600	4798	0		120643	20471	1846	120643	100	88	####### ####
36	2012	1468	550	600	4990	0		105089	18760	1506	105089	100	89	####### ####
37								TOTAL TAXES FOR 30 YRS		76037				

Table 2. Variation of sample retirement table.

39	B	C	D	E	F	G	H	I	J	K	L	M
40												
41												
42	1.04 SS INCREASE						FORMULA					
43	1.04 SPENDABLE INC. INCREASE						FOR IRA		TAX TABLE FOR COUPLE BOTH OVER 65			
44	1.11 IRA INCREASE						AMORTIS.			-1	0	
45	1.11 LIQUID ASSETS INCREASE						15			7500	11	
46							.209004			8500	121	
47							.790996			10000	296	
48							7.19087			11000	426	
49							32639			12000	564	
50										13000	714	
51										14000	864	
52										15000	1014	
53										16000	1149	
54										17000	1336	
55										18000	1506	
56										19000	1676	
57										20000	1846	
58										22000	2226	
59										24000	2600	
60										26000	3058	
61										28000	3518	
62										30000	4000	
63										32500	4600	
64										35000	5330	
65										37500	6100	
66										40000	6900	
67										45000	8600	
68										50000	10300	
69										55000	12300	
70										60000	14300	
71										65000	16400	
72										70000	18600	
73										75000	20800	
74										80000	23000	
75										85000	25200	
76										90000	27200	
77										95000	29800	
78										100000	32200	
79										105000	34400	
80										110000	36600	
81										115000	39500	
82										120000	42000	
83										125000	44500	
84										130000	47000	
85										135000	49500	
86										140000	52000	
87										145000	54500	

Table 3. Income tax table with constants.

monthly pension or other income. If there is none, this column simply shows zeros as in Table 1.

● Column F represents the whole reason for this exercise: spendable dollars after taxes. Annual increases to keep up with inflation can be varied by different multipliers. The increase is a straight percentage unless more money is desired in the early years, offset by correspondingly lower amounts later. F7 is the starting amount of personal budget needs, but, of course, it is related to available funds.

● Columns G and I contain the amounts of assets, tax-sheltered and liquid, as mentioned above. By entering different amounts in locations G7 and I7, individual financial conditions are accommodated, instead of the examples shown here.

● Column H indicates amounts withdrawn from the tax-sheltered rollover account. While no withdrawals are required until 1993 by the IRS, minimizing income taxes or special needs might indicate earlier payouts. From location H17 on, amounts are calculated by the program to amortize the fund from G16 over the required number of years. Additional withdrawals in this column have to be entered manually.

● Column J, taxable income, is calculated by adding the annual equivalents of columns E and H plus the income from the liquid assets in column I.

● Column K shows income taxes that are entered automatically from special tables, shown together with the constants in Table 3. This table is strictly for the technically minded. The Lookup Function of VisiCalc fills in the taxes. A tax schedule can easily be changed, as the law changes, and can be entered in with as much detail as needed. (When forecasting for a period of 30 years, super accuracy isn't possible.)

● Columns L, M, and N are added to give an overall picture of the financial retirement status in any year for which a future prediction has been made. Column L shows that net worth can be maintained with slightly more conservative spending, or be almost depleted by the time very old age is reached. At any rate, assets decline slowly (just as one gets older, requiring less). Column N shows the retiree's age for reference purposes.

● Columns O and P are simply a graphic display of assets for each year.

The program could be used by corporate planners or financial consultants wanting to show employees what to expect, financially, for their retirement years.

Despite inflation, retirement can be financially comfortable, with proper planning using this Apple II program application. ■

Losers I Have Known

From the many software products available for the Apple computer, here are a few you may not have heard of... and may never wish to hear of again.

by Chuck Doherty

Anyone involved, even casually, with the computer field is familiar with the more popular computer programs. VisiCalc and Adventure are two that come to mind immediately. Sadly, for every successful business program or computer game on the market, there are many more with less-than-optimum performance.

Since every program has behind it a proud author, it seems only fair to occasionally take a look at some of the "loser" programs available and bring them some much deserved, if not earned, attention.

Type-O-Rama Word Processor, by Ted's Package Store and Software Inc. The unique feature of Type-O-Rama is that it accurately simulates the use of an inexpensive manual typewriter. The computer's speaker emits a loud CLACK every time a key is depressed. Striking two keys without pausing between keystrokes can make two characters appear on the screen, one superimposed on the other. This also jams the keyboard, requiring you to lift the top cover of the computer and reset a small lever. A light pen is also included, which is used in the correction mode. To remove an unwanted character from the screen, vigorously rub it with the light pen until it gradually fades away (it will never completely disappear). There is no insertion mode and documents cannot be stored to disk.

Mrs. O'Leary's Spell-Check, from Mrs. O'Leary, Canton, OH. Somewhat unique in concept, this "program" requires you to mail printed documents to Mrs. O'Leary for correction. A junior high school English teacher for over forty years, Mrs. O'Leary knows how to spell just about everything. Just 6–8 weeks after receiving your document she will return it with all spelling errors circled with her famous red pencil. For a small additional fee, Mrs. O'Leary will point out errors in punctuation and sentence structure. No medical documents are allowed.

Backpack Companion, from L.L. BEEN Inc., Kinniebunkport, ME. Designed for the serious outdoorsman who is also a serious computer user, this program is made for computer use on hiking and camping trips. Unique physically, it is one of the few programs on the market shipped in a Thinsulate envelope rather than one made of paper. The program graphically displays illustrations and descriptions of various plants and small animals that may be safely eaten in the wild. Easy, one-key access is provided for emergency data retrieval, such as snake-bite first aid and frostbite remedies. The program contains a limited word processing routine intended for the writing of a last will and testament, should all else fail. The extension cord is not included.

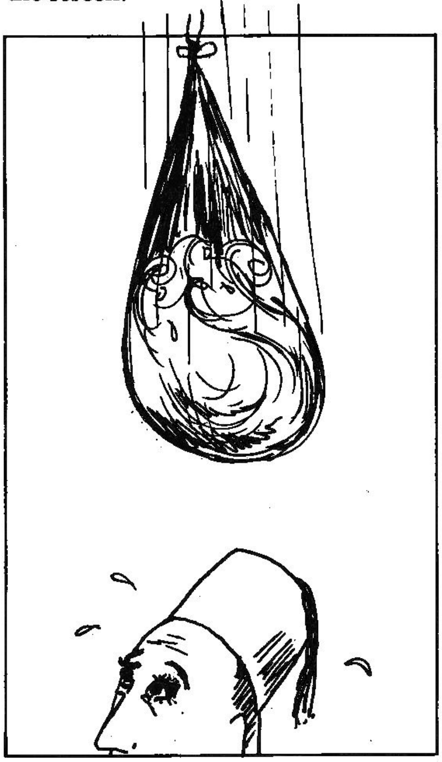
Sort-of-Sort, R.P. McMurphy Software Inc., Bellevue, NY. Sorting routines are nothing new, but one like this doesn't come along every day. Used with a large database file, this program first randomizes the file and then sorts it according to completely arbitrary and changeable criteria. For instance, a large mailing list might be sorted by the number of vowels in the street name, while another version of the program could sort the same list according to the difficulty of saying the customer's middle name three times fast. Mr. McMurphy is available during visiting hours on Thursday afternoons from 2 till 4 to answer questions on program applications. He can be found in Building Seven, Ward G.

Career Selection Program, from the U.S. Department of Defense. Aimed at young people trying to select the best field to enter in life, this program asks questions determining aptitude for various careers and future education. Although several similar programs exist, this one stands apart since, regardless of the answers given to the questions or any of the data the user furnishes, the

Address correspondence to W. Charles Doherty, 32 Meadowood Drive, South Dartmouth, MA 02748.

recommended career is always enlistment in the United States Army.

Shriners from Space, by Tirebiter Software Inc. This variation of the popular arcade game pits you against an unending parade of grown men driving teeny-tiny Thunderbirds down a city street. The object of the game is to drop paper bags filled with water from your hotel room onto the vehicles as they pass. Watch out for the clowns on tricycles; if you hit one, you get a pie in the screen.



Teacher's Helper, by Blackboard Jungle Software Inc. Finally, an education program for difficult students. This program demands respect with no bandying about. Questions are asked and the student responds correctly or pays the price. The software package comes with an interface board and a set of clip-on electrodes to reinforce the importance of paying attention. Teacher's helper is not recommended for use with persons wearing metal studded clothing.

Space does not permit mention of the other fine "loser" programs available today. Perhaps an annual award for the most outstanding entry in the field is in order. Any suggestions for nominees are welcome. ■

Star Dust

For everyone who creates games, or wants to, this asteroid generator will be invaluable.

by Dave Schroeder

Imagine you're the captain of the USS *Apple*, on a mission into deep space and warp out, factor 4. ZOOM! The stars pick up speed and fly past your ship. "Great," you say, but you only have Applesoft and it's too slow to display 100 to 200 moving stars.

I have often played Space Invaders, Star Cruiser, and Apple Galaxian, only to marvel at the speed of the graphics displays. The average user should have small subroutines to spruce up a program and show professional displays.

I attempted to do this using 6502 machine code to display an asteroid belt that speeds past the viewer by rolling around the available screen. This proved to be a formidable challenge, but I hurdled most of the barriers.

Once I got the stars moving, a speed control was added by reading the paddle (0) for a value, from 0–255. The function INT(x/16) + 1 was used as the change in displacement of the stars moving down the screen. Also, a routine was added to check when the paddle button is depressed. The routine then jumps back to the monitor. (See Star Dust, Listing 1, a–c.)

After a little playing and restructuring, this subroutine could be used in any space adventure program by using the Call(x) function. I have also included a program, Star Helper (Listing 2), that will generate the random star field and call Star Dust. Check it out and have fun. ■

Address correspondence to David J. Schroeder, 34 Williamstowne Ct., Apt. 6, Cheektowaga, NY 14227.

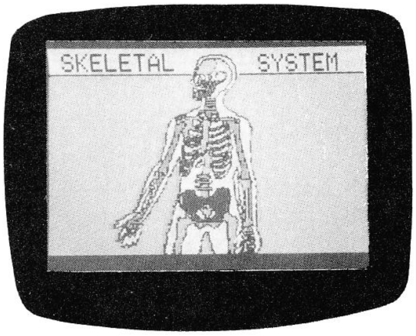
```
0300:           ;
0300:           ;
0300:           ;  STAR DUST
0300:           ;
0300:           ;      BY
0300:           ;
0300:           ;DAVID SCHROEDER
0300:           ;
0300:           ;
0300:           ;THIS PROGRAM ALLOWS THE
0300:           ;USER TO CREATE AND EFFECT
0300:           ;OF MOVING ASTEROIDS AS
0300:           ;PART OF THE BACKGROUND FOR
0300:           ;A PROGRAM. THIS PROGRAM
0300:           ;WILL LOOP TILL PDL 0 IS
0300:           ;PRESSED (EXITING INTO THE
0300:           ;MONITOR). THE SPEED CAN
0300:           ;BE CHANGE BY MOVING THE
0300:           ;PADDLE OR BY CHANGING THE
0300:           ;DELTA Y LOCATION ($08).
0300:           ;
0300:           ;
0000:            BLACK   EQU  $00
0008:            DELTA.Y EQU  $08
00E4:            HCOLOR  EQU  $E4
F3E2:            HGR     EQU  $F3E2
F457:            HPLOT   EQU  $F457
00A0:            LINES   EQU  $A0
FF69:            MONITOR EQU  $FF69
C061:            PDBUTO  EQU  $C061
FB1E:            PREAD   EQU  $FB1E
0006:            STRPTR  EQU  $06
1000:            STRTBL  EQU  $1000
00FF:            WHITE   EQU  $FF
0007:            YROLL   EQU  $07
0300:           ;
0300:           ;CLEAR THE GRAPHICS PAGE
0300:           ;AND SET UP INITIAL POINTERS.
0300:           ;
0300:20 E2 F3            JSR  HGR
0303:A9 00               LDA  #$00
0305:85 06               STA  STRPTR
0307:85 07               STA  YROLL
0309:85 08               STA  DELTA.Y
```

Listing 1a. Setting up the asteroid backround.

FOR COMPLETE GRAPHICS:

VersaWriter

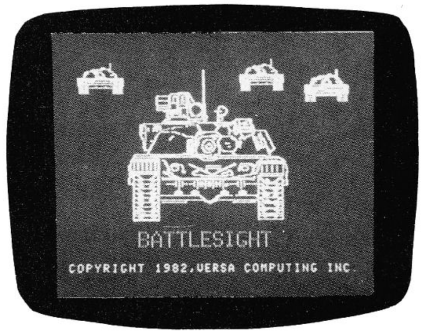
EDUCATION

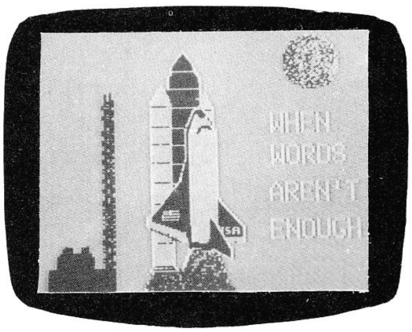


ARTIST

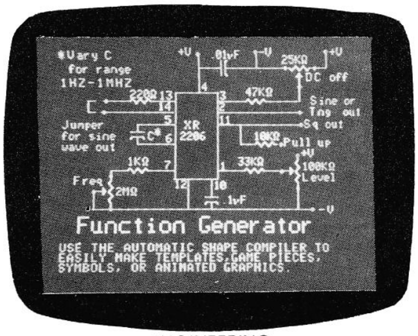


GAME PROGRAMMER

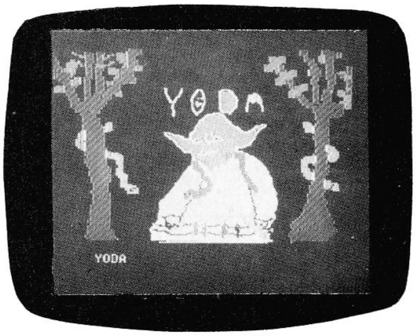




HOBBIEST

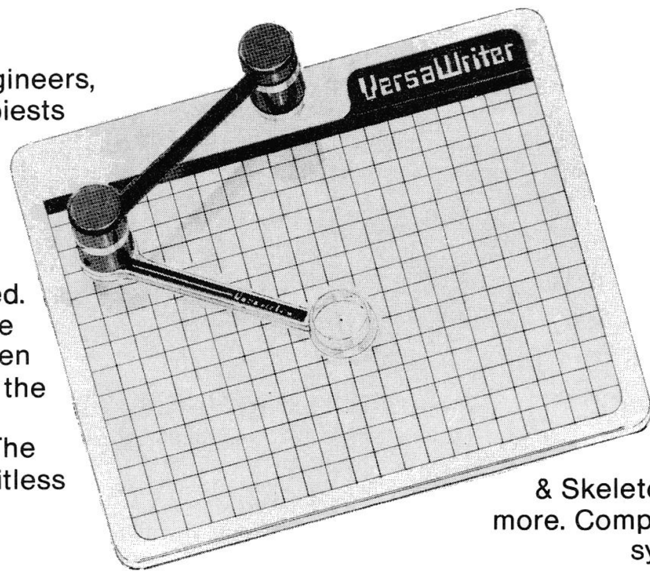


ENGINEERING



CHILDREN

Teachers, artists, engineers, programmers & hobbiests find VersaWriter an easy to use tool for creating micro computer graphics. No programming experience is required. Pictures can be made by simply tracing. Even children can explore the exciting world of computer graphics. The VersaWriter is as limitless as your imagination.



VersaWriter contains complete software for drawing with color, brushes & dots. Add text or fill in over 100 colors. Create your own shapes and place anywhere on the screen. Use Area/Distance, Move Picture, Electronic Drawing & Skeleton programs plus much more. Complete hardware/software system for Apple II/II+/IIe - **$299.00**

Versa Computing Products are available at your local computer products store.

Distributed by:

VERSA COMPUTING, INC.

Computerland Corp.
Hayward, Calif.

Softsel Computer Products
Inglewood, Calif.

Pete & Pam Computers
Lancashire, England

Micron Distributing
Toronto, Canada

Program Spektrum
Bromma, Sweden

Micro Products Sales Group
Lynn, Mass.

Educational Media
Washington, Penn.

ESD Laboratories
Tokyo, Japan

Blue Ridge Computers
Capetown, South Africa

3541 Old Conejo Road, Suite 104 • Newbury Park, CA 91320 • (805) 498-1956

Circle 239 on Reader Service card.

```
030B:           ;
030B:           ;READS POINTS FROM STRTBL
030B:           ;WHEN 1ST CO-ORD IS 0 THEN
030B:           ;YOU ARE AT END OF TABLE.
030B:           ;
030B:A6 06      START    LDX   STRPTR
030D:BD 00 10            LDA   STRTBL,X
0310:D0 22               BNE   MOVE
0312:85 06               STA   STRPTR
0314:A5 07               LDA   YROLL
0316:18                  CLC
0317:65 08               ADC   DELTA.Y
0319:C9 A0               CMP   #LINES
031B:90 02               BCC   START1
031D:E9 A0               SBC   #LINES
031F:85 07      START1   STA   YROLL
0321:A2 00               LDX   #$00
0323:           ;
0323:           ;RE-READ PADDLE 0 FOR NEW
0323:           ;DELTA Y. THIS CAN BE
0323:           ;CHANGED FOR USER PROGRAM
0323:           ;CONTROL.
0323:           ;
0323:20 1E FB            JSR   PREAD
0326:98                  TYA
0327:29 F0               AND   #$F0
0329:18                  CLC
032A:6A                  ROR   A
032B:6A                  ROR   A
032C:6A                  ROR   A
032D:6A                  ROR   A
032E:69 01               ADC   #$01
0330:85 08               STA   DELTA.Y
0332:90 D7               BCC   START
```

Listing 1b. Star Table.


```
0334:           ;
0334:           ;FIND THE OLD POINT AND
0334:           ;TURN IT OFF - PLOT BLACK
0334:           ;
0334:48         MOVE     PHA
0335:E8                  INX
0336:BD 00 10            LDA   STRTBL,X
0339:18                  CLC
033A:65 07               ADC   YROLL
033C:90 02               BCC   MOVE1
033E:E9 A0               SBC   #LINES
0340:C9 A0      MOVE1    CMP   #LINES
0342:90 02               BCC   MOVE2
0344:E9 A0               SBC   #LINES
0346:A8         MOVE2    TAY
0347:68                  PLA
0348:48                  PHA
0349:AA                  TAX
034A:98                  TYA
034B:48                  PHA
034C:A0 00               LDY   #BLACK
034E:84 E4               STY   HCOLOR
0350:20 57 F4            JSR   HPLOT
0353:           ;
0353:           ;FIND NEW POINT BY OFFSET
0353:           ;AND PLOT IT WHITE
0353:           ;
0353:68                  PLA
0354:18                  CLC
0355:65 08               ADC   DELTA.Y
0357:C9 A0               CMP   #LINES
0359:90 02               BCC   MOVE3
035B:E9 A0               SBC   #LINES
035D:A8         MOVE3    TAY
035E:68                  PLA
035F:AA                  TAX
0360:98                  TYA
0361:A0 FF               LDY   #WHITE
0363:84 E4               STY   HCOLOR
0365:A0 00               LDY   #$00
0367:20 57 F4            JSR   HPLOT
036A:E6 06               INC   STRPTR
036C:E6 06               INC   STRPTR
036E:           ;
036E:           ;IF PADDLE BUTTON 0 IS
036E:           ;PRESSED THE EXIT TO
036E:           ;THE MONITOR ELSE DO
036E:           ;ANOTHER STAR.
036E:A9 80               LDA   #$80
0370:2C 61 C0            BIT   PDBUT0
0373:30 02               BMI   EXIT
0375:10 94               BPL   START
0377:4C 69 FF   EXIT     JMP   MONITOR

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

Listing 1c. Move a Star.

```
5  HOME : VTAB 21: PRINT CHR$ (4); "BLOAD STAR DUST"
10 FOR P = 4096 TO 4166 STEP 2
20 POKE P, INT (RND (1) * 255) + 1
30 POKE P + 1, INT (RND (1) * 150) + 1
40 NEXT P
50 POKE 4168,0
60 CALL 768
```

Listing 2. Star Helper.

EVERYONE NEEDS A . . .



SAVE YOUR DATA FROM POWER OUTAGES!

BACKUP FOR YOUR COMPUTER, MONITOR, PRINTER AND 5¼″ FLOPPY AND HARD DISC DRIVE

• Automatically stops annoying problems from power line interruptions and brown outs • You need standby power to save data • Maintenance free backup power available in 115 volt or 220 volt • 50 or 60 HZ • 150 watts • Complete versatility — operate your system from a 12 volt source, i.e., automobile cigarette lighter, boat or airplane • Rugged self contained gel cell battery • No voiding warranty — no cutting wires • Automatic audio alarm warning tone during commercial power failure or interrupt • UL listed • FCC approved • Transient voltage suppressor gives added insurance from line voltage spikes, utilizing Zener Ray™ • Green/red LED power status indicator • Green — normal AC line power • Slow blinking red — at least 6 minutes of remaining standby power • Fast blinking red — approximately 2 minutes of remaining battery power • Solid state technology unexcelled by any UPS power unit in its class.

RH ELECTRONICS, INC.

COPYRIGHT © 1981 - PATENTS PENDING

566 IRELAN, BUELLTON, CA 93427

(805) 688-2047

SEE YOUR RH ELECTRONICS PRODUCTS DEALER

FOR YOUR APPLE II*:

SUPER FAN II™	$ 74.95
SUPER FAN II™/ZENER RAY™	$109.00
SUPER RAM II™	$125.00
RH 12 VOLT TRANSVERTER	$149.00

FOR MICRO COMPUTERS:

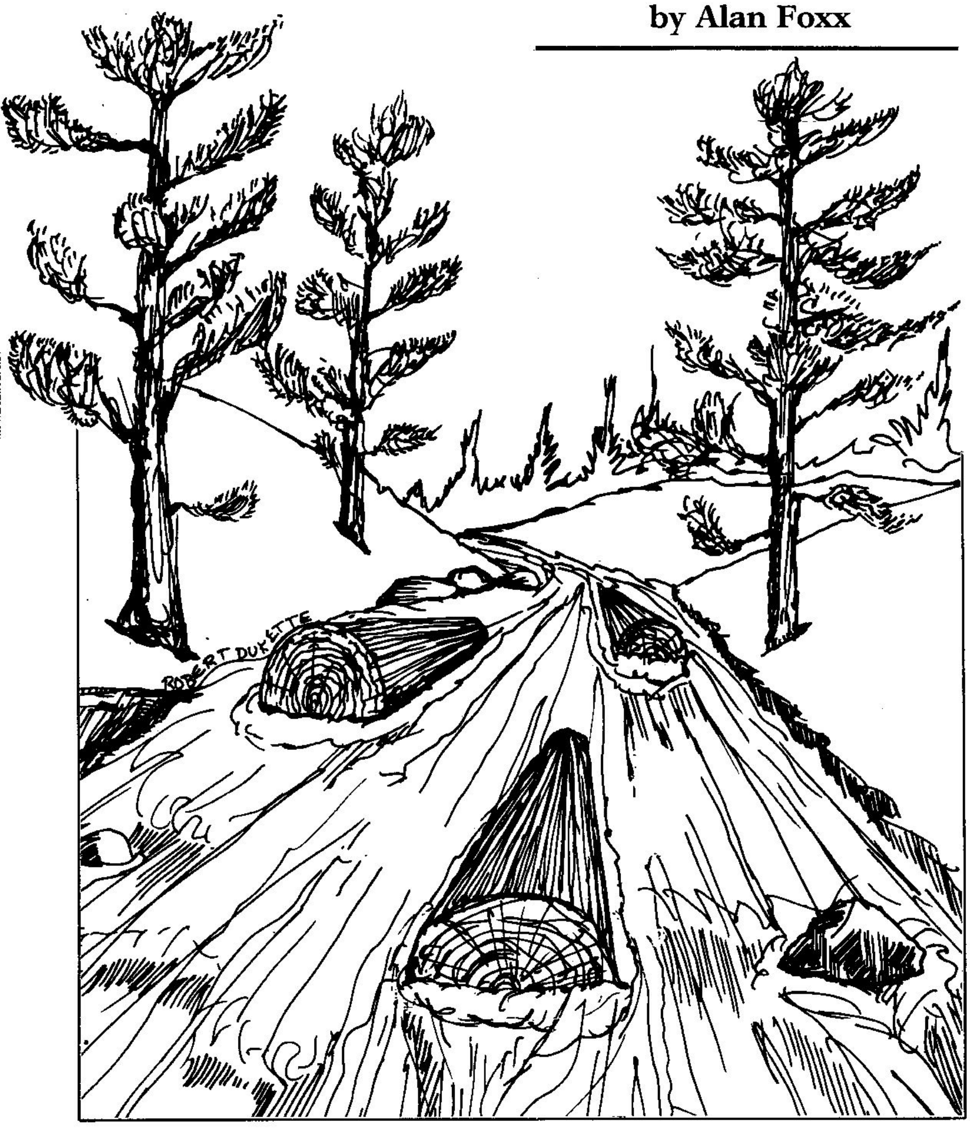
GUARDIAN ANGEL™	$595.00

Circle 91 on Reader Service card.

Logger

Get in the swim with this action game to test your skill and stamina.

by Alan Foxx



For all you gamesters, here's an interesting variation on a popular theme. As you swim downstream, drawn inexorably by a fierce current, you'll pass by markers that represent money bags. By manipulating your swimmer with the cursor arrows or game paddles, you can gather up the bags and score lots of points.

But watch out for the logs! In the real world, logs flow predictably with the stream. But in Logger you'll have to contend with eddies and crosscurrents that set logs across your course from every which way. These logs weren't designed to give a weary swimmer a breather—they'll knock you out of the game if you fail to dodge them.

As your skill and stamina improve, scoring gets harder, but the potential for satisfying your greed multiplies. More $$$! The high score and average score are displayed at the end of each game.

Logger is a low-resolution graphics game with frankly gratuitous sound. You might want to modify the game to suit yourself—but here's the basic code to get you started. ■

Address correspondence to Alan Foxx, 28090 Tavistock Trail, Southfield, MI 48034.

Program listing. Logger.

```
10  REM   ************
15  REM  *   LOGGER   *
18  REM  * WRITTEN BY *
25  REM  * ALAN  FOXX *
28  REM  * 12/15/1982 *
30  REM   ************
31 :
32  POKE 768,173: POKE 769,48: POKE
     770,192: POKE 771,136: POKE
     772,208: POKE 773,4: POKE 77
     4,198: POKE 775,7
35  POKE 776,240: POKE 777,8: POKE
     778,202: POKE 779,208: POKE
     780,246: POKE 781,166: POKE
     782,6: POKE 783,76: POKE 784
     ,0: POKE 785,03: POKE 786,96

40 GA = 0:HS = 0:GT = 0
50  TEXT : HOME : VTAB (11): HTAB
     (16): FLASH : PRINT "LOGGER"
     : NORMAL : VTAB (13): HTAB (
     10): PRINT "WRITTEN BY ALAN
     FOXX"
60  FOR D = 1 TO 200: NEXT D: GOSUB
     1080: FOR D = 1 TO 500: NEXT
     D
70  TEXT : HOME : PRINT "WELCOME
     TO LOGGER.  DO YOU PREFER GA
     ME": PRINT "PADDLES OR THE K
     EYBOARD? (K/P) =>";: GET A$:
      PRINT A$
80  IF A$ = "K" THEN CN$ = "K": GOTO
     110
90  IF A$ = "P" THEN CN$ = "P": GOTO
     110
100  GOTO 70
110  FOR D = 1 TO 990: NEXT D: POKE
      - 16368,0:TURNS = 3
120  FOR I = 1 TO 20: POKE 6,244:
      POKE 7,1: CALL 768
130  POKE  - 16368,0
140  TEXT : HOME : VTAB (4): PRINT
     "LOGGER:": VTAB (6): PRINT "
       THE OBJECT OF THE GAME IS
     TO GRAB AS": PRINT "MANY DOL
     LAR BAGS, REPRESENTED AS GRE
     EN": PRINT "DOTS, AS YOU CAN
      BEFORE EITHER TIME"
150  IF CN$ = "K" THEN  PRINT "RU
     NS OUT OR ONE OF THE LOGS GE
     TS YOU.": PRINT "YOU WILL BE
      IN CONSTANT FORWARD MOTION.
     ": PRINT "TO STEER LEFT OR R
     IGHT, USE THE ARROW": PRINT
     "KEYS. ";: FLASH : PRINT "TO
      GO STRAIGHT, PRESS THE"
160  IF CN$ = "K" THEN  PRINT "SP
     ACE BAR.": NORMAL : GOTO 210

170  PRINT "RUNS OUT OR ONE OF TH
     E LOGS GETS YOU.": PRINT "YO
     U WILL BE IN CONSTANT FORWAR
     D MOTION.": PRINT "TO STEER
     LEFT OR RIGHT, USE THE GAME"
     : PRINT "PADDLE.  WHICH PADD
     LE DO YOU WANT TO": PRINT "U
     SE: PADDLE #0 OR PADDLE #1?
      TYPE IN A"
180  PRINT "'0' OR A '1'.  ==>";:
      GET A$:P =  VAL (A$): IF P >
     1 THEN 140
190  PRINT P
200  PRINT : PRINT "BONUS TURN EV
     ERY FIVE LEVELS."
210  PRINT : PRINT "GREEN BAGS AR
     E WORTH 15 POINTS EACH.": GOSUB
     1060: GOSUB 1060: GOSUB 1060
     : GOSUB 1060: IF CN$ = "K" THEN
      PRINT : INPUT "PRESS RETURN
      TO GO ON. =>";A$: PRINT
220  POKE  - 16368,0:LV = LV + 1:
     CG = 0: IF  INT (LV / 5) * 5
      = LV THEN TURN = TURN + 1: HOME
     : FLASH : PRINT "YOU HAVE EA
     RNED AN EXTRA TURN!!!!!!!!!"
     : NORMAL : GOSUB 1080
230 TJ = 0.129
240  GR : IF LV > 4 THEN TJ = 0.0
     85
250  COLOR= 15: FOR I = 0 TO 39: HLIN
     0,39 AT I: NEXT I
260  COLOR= 0
270  GOSUB 280: GOTO 380
280 X3 = 5:X4 = 11:X5 = 25:X6 = 3
     1
290  COLOR= 15: HLIN 1,38 AT 5: HLIN
     1,38 AT 25: HLIN 1,38 AT 15:
      HLIN 1,38 AT 35: VLIN 1,38 AT
     15: VLIN 1,38 AT 25
300  FOR C5 = 1 TO 6
310  IF  INT (LV / 2) * 2 <  > LV
      OR LV > 4 THEN  COLOR= 0: PLOT
     X4,5: PLOT X6,25: PLOT 39 -
     X4,15: PLOT 39 - X6,35: COLOR=
     15: PLOT X3,5: PLOT X5,25: PLOT
     39 - X3,15: PLOT 39 - X5,35
320  IF  INT (LV / 2) * 2 = LV OR
     LV > 4 THEN  COLOR= 0: PLOT
     X4,5: PLOT 25,X6: PLOT 15,39
      - X4: PLOT 39 - X6,35: COLOR=
     15: PLOT X3,5: PLOT 25,X5: PLOT
     15,39 - X3: PLOT 39 - X5,35
330 X4 = X4 + 1: IF X4 > 38 THEN
     X4 = 2
335  POKE 6,255 - (C5 * 10): POKE
     7,3: CALL 768
340 X3 = X3 + 1: IF X3 > 38 THEN
     X3 = 2
350 X5 = X5 + 1: IF X5 > 38 THEN
     X5 = 2
360 X6 = X6 + 1: IF X6 > 38 THEN
     X6 = 2
370  NEXT C5: RETURN
380 U2 =  INT (LV * 2.5)
390  COLOR= 14: FOR U0 = 1 TO U2
400 GY =  INT ( RND (1) * 34) + 3
     : IF GY = 5 OR GY = 15 OR GY
      = 25 OR GY = 35 THEN 400
410 GX =  INT ( RND (1) * 33) + 4
     : IF GX = 15 OR GX = 25 THEN
     410
420  IF  SCRN( GX,GY) = 14 THEN 4
     00
430  PLOT GX,GY: NEXT U0
440  COLOR= 1: VLIN 3,36 AT 0: VLIN
     3,36 AT 39
450  HOME
460  REM  TIME
470  GOSUB 480: GOTO 590
480  COLOR= 2: HLIN 0,39 AT 39
490  COLOR= 13: PLOT 19,38:X = 19
     :Y = 38
500  IF LV < 2 THEN  VTAB (23): PRINT
     "NOW REMEMBER, DON'T BUMP IN
```

Listing continued.

Listing continued.

```
     TO THE LOGS": PRINT "OR THE
     RED SIDES!!!": FOR D = 1 TO
     2200: NEXT D
510  HOME : PRINT "^ TIME ^"; TAB(
     17);"SCORE"; TAB( 32);"LEVEL
      ";LV
520  VTAB (23): PRINT "ON YOUR MA
     RKS..."
530  GOSUB 1060: VTAB (23): PRINT
     "GET SET...              "
540  GOSUB 1060: VTAB (23): POKE
     6,175: POKE 7,7: CALL 768
550  PRINT "GO!!!              "; TAB(
     18);SC; TAB( 23);"TURNS LEFT
      -> ";TURNS:HW$ = "H"
570 TIME = 39.9: POKE  - 16368,0
580  RETURN
590 WH$ = " "
600  IF  INT (LV / 2) * 2 <  > LV
      OR LV > 4 THEN  COLOR= 0: PLOT
     X4,5: PLOT X6,25: PLOT 39 -
     X4,15: PLOT 39 - X6,35: COLOR=
     15: PLOT X3,5: PLOT X5,25: PLOT
     39 - X3,15: PLOT 39 - X5,35
610  IF  INT (LV / 2) * 2 = LV OR
     LV > 4 THEN  COLOR= 0: PLOT
     X4,5: PLOT 25,X6: PLOT 15,39
      - X4: PLOT 39 - X6,35: COLOR=
     15: PLOT X3,5: PLOT 25,X5: PLOT
     15,39 - X3: PLOT 39 - X5,35
620 X4 = X4 + 1: IF X4 > 38 THEN
     X4 = 2
630 X3 = X3 + 1: IF X3 > 38 THEN
     X3 = 2
640 X5 = X5 + 1: IF X5 > 38 THEN
     X5 = 2
650 X6 = X6 + 1: IF X6 > 38 THEN
     X6 = 2
660  GOSUB 880
670  IF CN$ = "P" THEN 710
680  IF  PEEK ( - 16384) = 149 THEN
      GOSUB 940
690  IF  PEEK ( - 16384) = 136 THEN
      GOSUB 1000
700  GOTO 730
710  IF  PDL (P) > 170 THEN  GOSUB
     940
720  IF  PDL (P) < 85 THEN  GOSUB
     1000
730 TIME = TIME - TJ: COLOR= 15: PLOT
     TIME,39
740  IF TIME < 1 THEN HW$ = "T": GOTO
     790
750  IF WH$ = "L" THEN HW$ = "H":
      GOTO 790
760  IF CG =  > U2 THEN 860
770  FOR D = 1 TO UB: NEXT D
780  GOTO 590
790  COLOR= 15: PLOT X,Y: GOSUB 1
     270: FLASH : IF HW$ = "H" THEN
      PRINT "YOU GOT HIT."
800  IF HW$ = "T" THEN  PRINT "TI
     ME RAN OUT."
810  NORMAL : GOSUB 1060: GOSUB 1
     060
820 TURNS = TURNS - 1
830  IF TURNS <  = 0 THEN  GOSUB
     1100
840  IF TURNS > 0 THEN  GOSUB 280
     : GOSUB 480: GOTO 590
850  GOTO 70
860  HOME : VTAB (23): FLASH : PRINT
     "GREAT JOB!!!   NOW ONTO THE
      NEXT LEVEL.": NORMAL : GOSUB
     1060: GOSUB 1060: GOTO 220
870  END
880  REM  ^UP^
890  IF Y - 1 < 0 THEN  COLOR= 13
     : PLOT X,38: COLOR= 15: PLOT
     X,Y:Y = 38: RETURN
900  IF  SCRN( X,Y - 1) <  = 1 THEN
     WH$ = "L": RETURN
```

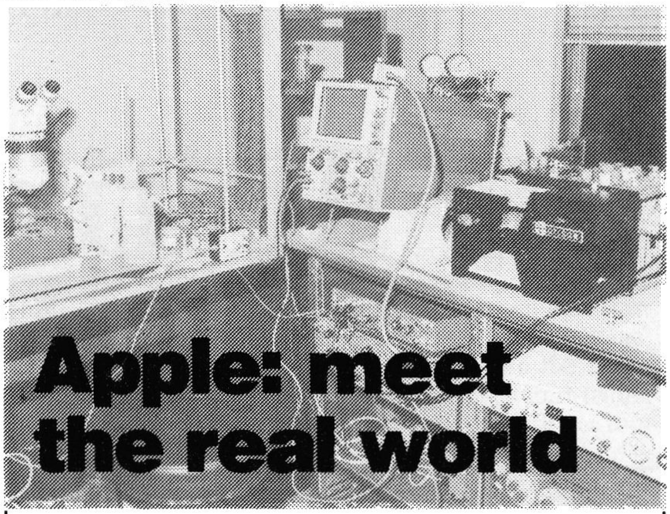
Listing continued.

Listing continued.

```
910  IF  SCRN( X,Y - 1) = 14 THEN
      GOSUB 1300
920  COLOR= 13: PLOT X,Y - 1
930  COLOR= 15: PLOT X,Y:Y = Y -
     1: RETURN
940  REM  -> LEFT ->
950  IF X + 1 > 39 THEN  RETURN
960  IF  SCRN( X + 1,Y) <  = 1 THEN
     WH$ = "L": RETURN
970  IF  SCRN( X + 1,Y) = 14 THEN
      GOSUB 1300
980  COLOR= 13: PLOT X + 1,Y
990  COLOR= 15: PLOT X,Y:X = X +
     1: RETURN
1000  REM  <- RIGHT <-
1010  IF X - 1 < 0 THEN  RETURN
1020  IF  SCRN( X - 1,Y) <  = 1 THEN
     WH$ = "L": RETURN
1030  IF  SCRN( X - 1,Y) = 14 THEN
      GOSUB 1300
1040  COLOR= 13: PLOT X - 1,Y
1050  COLOR= 15: PLOT X,Y:X = X -
     1: RETURN
1060  FOR D = 1 TO 800: NEXT D: RETURN

1080  FOR S = 1 TO 2
1082  IF S = 1 THEN ST = 255:EN =
     1:INC =  - 5
1083  IF S = 2 THEN EN = 255:ST =
     1:INC = 5
1085  FOR I = ST TO EN STEP INC: POKE
     6,I: POKE 7,4: CALL 768: POKE
     6,I / 2: POKE 7,4: CALL 768:
      POKE 6,I / 3: POKE 7,4: CALL
     768: POKE 6,I / 4: POKE 7,4:
      CALL 768: POKE 6,255 - I: POKE
     7,4: CALL 768: NEXT I
1087  NEXT S
1090  RETURN
1100  REM  SCOREBOARD
1110  TEXT : HOME : VTAB (2): HTAB
     (7): PRINT "YOUR CURRENT SCO
     REBOARD IS:"
1120  IF SC > HS THEN HS = SC
1130 GT = GT + SC
1140 GA = GA + 1:PCNT =  INT (GT /
     GA)
1150  VTAB (5): PRINT  TAB( 6);"H
     I SCORE"; TAB( 25);"SCORE"
1160  VTAB (6): HTAB (6): PRINT "
     ---------": VTAB (6): HTAB (2
     5): PRINT "-----"
1170  VTAB (7): PRINT  TAB( 9);HS
     ; TAB( 26);SC
1180  VTAB (12): PRINT  TAB( 6);"
     GAMES PLAYED"; TAB( 20);"AVE
     RAGE SCORE"
1190  VTAB (13): PRINT  TAB( 6);"
     ------------"; TAB( 20);"---
     ----------"
1200  VTAB (14): PRINT  TAB( 11);
     GA; TAB( 20);PC;" PER GAME."

1205  GOSUB 1080
1210  POKE  - 16368,0
1220  VTAB (22): INPUT "WANT TO P
     LAY LOGGER AGAIN? (Y/N) =>";
     A$
1230 LV = 0:SC = 0
1240  IF  LEFT$ (A$,1) = "Y" THEN
      RETURN
1250  IF  LEFT$ (A$,1) = "N" THEN
      PRINT : PRINT "BYE": END
1260  GOTO 1220
1270  FOR I = 255 TO 1 STEP  - 5:
      POKE 6,I: POKE 7,1: CALL 76
     8: NEXT I: RETURN
1300 SC = SC + 15:CG = CG + 1: VTAB
     (23): HTAB (18): PRINT SC
1310 B =  PEEK ( - 16336) +  PEEK
     ( - 16336): RETURN
```

Writing for Readers

Newcomers to computing need lots of help to get started. Why must documentation be written by and for engineers?

by Fred Huntington

If someone out there is interested in making a lot of money, I've got just the idea for you. All you have to do is write readable instruction manuals. Nowhere in all of history have there been worse instructions for anything than there are for computer products—and especially printers.

I like to use my father-in-law as an example. He is a professional writer who wanted absolutely nothing to do with computers. I convinced him that it would be great to have an Apple at home to write his novel (forthcoming now for twenty years). Now he uses the computer at work for all of his word processing.

After playing Little Brick-Out and Beer Run for two months, he decided it was time to get serious with word processing. So, I gave him the best word processor, the finest hardware I could find, and told him to go to it.

That was two months ago, and the system still isn't running. My father has tried, and tried, and tried to be on-line. I finally determined a hardware problem was causing the foul-up.

The point is that there was no way a novice could set up the printer, the correct cards, and the word processor, without a great deal of help.

One of the great things about the Apple is that most manuals are written so that a complete novice can open a package and be programming in a matter of minutes. Why can't other manufacturers see the success of this formula?

There are a lot more people out there like my father-in-law who really don't care how the computer functions. They just want the end product.

"After playing Little Brick-Out and Beer Run for two months, he decided to get serious."

My father-in-law is a great writer, but a mechanic he is not. The box the printer comes in should have directions for him on the outside telling how to open the box without injuring himself, and it should specify what tools are needed to open the box.

Once the box is open, there should be a sheet on top that explains how to unpack it. Next a manual for the printer you have in front of you should be found, describing all of your printer's options. None of this, "If you have model #Z143-236HBAD, then you must insert the floomer carefully between the twixits (unless your serial number begins with a 3 or a 4, in which case you skip step 7b). If your model is different, then the floomer must be tangent to the twikle (if using an RS-232 interface). If you have the optional form-flatner, then twist 90 degrees before inserting, careful not to bend the twikle. If you have the pin speeder, and model #Z143-236HBAD, then see your dealer before proceeding."

Now the printer is out of the box and sitting next to the computer. Here are the instructions that meet the user (these are the actual instructions that accompanied one of the best printers on the market today!):

"Selection of serial interface configuration strapping to meet the user's particular installation is accomplished by installing jumpers into a dip strapping platform (Z2) on the controller circuit board, next to interface connector J1."

I don't understand all of that garbage. How is a first-time user supposed to know what to do?

Here's another quote from one more famous printer manufacturer regarding interfaces:

"A 40-pin printed circuit edge card connector located at the left rear of the printer provides the means for connecting the printer to the host device.

Address correspondence to Fred Huntington, PO Box 787, Corcoran, CA 93212.

A mating connector for preparing a customer supplied interface cable may be procured from -----. Two alternate connectors are 3M (P/N 3464-001) and T & B Ansley (P/N 609-4015M). The physical and electrical characteristics, pin orientation, and connector pin-outs of the printer interface connector are shown below."

Does anybody who makes the hardware and prints the documentation realize that there are computer owners who don't know, or don't care, what an interface is? Most users simply want to turn on the computer and have it work.

Some of you are going to scream and yell, "That's what you have dealers for. They'll show you." Well, you've been going to different dealers than I'm accustomed to.

A large number of computer users don't have nearby dealers. And a lot of computer tinkering is done at night when most dealers aren't open. Regardless, they want answers *now*, not tomorrow or the next day.

Okay, so we finally get the printer out of the way. We've messed around long enough so that we have the right interface card, the right cable (not a small matter) and we know where to plug it in.

Now comes the task of loading the paper. Thankfully, on some printers it's an easy task—whether you read the directions or not.

My favorite task is trying to configure the word processor for the make and model of printer and interface card I have installed.

The word processor will ask you which 80-column board, shift-key modification, and so on, are in your computer. Then it asks for the type of interface card and printer, whether you want auto line feed, parity, baud rate, your wife's maiden name and birth date. . . and anything else that seems totally irrelevant in getting your printer and word processor to work. (For this I gave up my Correcting Selectric II?)

Some word processors are better than others, but most are not written for the novice. I hate those that start off telling you, in first-grade language, how to insert the disk into the drive, and then on the next page start telling you how to write your own driver routine in machine language.

I'd go so far as to say that even the "good" documentation today is not good. It's going to have to be good enough so that my father-in-law, or my secretary, can operate the product the first time with confidence. In addition, a quick reference chart for those (like myself) who never read directions is a must!

Once I installed a new faucet in my house. It even came with a sound recording that told me how to accomplish the task. Maybe the computer industry can learn something from plumbers. ■

Con Job

Maybe those hotshots at UCSD and Microsoft have the right answers— but have they been asking the right questions?

by Paul Raymer

The other day I was reading Rodnay Zaks' "Handbook" on CP/M and reached page 247 when I realized that I would never be able to discover whodunnit, but it was very likely *not* the butler.

Because the text was so limited in scope I set the book down—to resume again another day—and browsed through my computer magazines and realized where the big money was—in Arcade Games!

Knowing that I was totally inept at creating hi-res graphics, inventing strange monsters and outer space beings, and deciding new ways to kill, maim, destroy or annihilate people, places and things, I decided to look up the *second* best way to make megabucks.

Disk utilities was the obvious answer!

This market is virtually untapped by the professional programmer and large software developers because of the increasing demand for new variations on Tic Tac Toe, Star Wars-type adventures and checkbook balancers, plus new twists on playing Old Maid, Pin the Tail on the Donkey and Towers of Hanoi and/or Saigon.

A quick review of the industry revealed that most of the Apple disk (or diskette as those folks in California still like to call them) programs were keyed around the concept of changing DOS to make the computer do things it was never meant to do. I think Mr. Wozniak is rich enough that I need not argue with him about what he had in mind, so I decided to leave the disk alone and just change the SCREEN so it did all sorts of neat things. After all, I don't even know who Mr. Sanyo is, and have never even heard of the inventor of the Zenith monitor, so surely they cannot object.

The CON:DOS system covers most of the things that can be done with an Apple computer, one disk drive, whatever memory you happen to have and a few weekends to shoot down trying to read the tiny print they use in this publication. Of course when the program is up and running it will be worth all the time you have spent typing it, correcting your many errors, and debugging the program (which should have been done by the author in the first place).

The name of the program was not a capricious choice. It was originally named WEIR DOS, after Herman Weir, the inventor of the expression ?SYNTAX ERROR, which Bill Gates so cleverly inserted into Applesoft Basic. I considered an alternate name, TUXE DOS after the Tuxe Indians, who discovered that Hex was not necessary to populate the State of Nevada, raise crops, build tepees or shoot Gila monsters or craps.

It was after careful consideration of the state of the art (and the economy) that the more contemporary name of CON:DOS was selected. The name means "C"omputer "O"n "N"ow, which is the only way to get this system to work properly. The name MIKA DOS was not used, since a patent search revealed that the Epson Company is making something that will work better than anyone else's, and MIKA DOS will be used to name that product. Probably.

CON:DOS: How It Works

The CON:DOS system consists of 15 separate programs, none of which are prepared in fast-moving assembly language. All are written in Basic, and you know who has to type all of the stuff in.

The programs are generally benignly short and easy to comprehend, type and use.

While the basic purpose of this series of programs is to provide a whole new world of thrills and excitement, a more mundane purpose may be to see how things work on your computer and how you can use similar programming techniques to create programs of your own. Any money you make in that

Address correspondence to Paul Raymer, c/o Paul's Electric Computer, Box 42813, Las Vegas, NV 89116

MENU

```
100  TEXT : HOME : CLEAR
110  REM

===============
MENU FOR CON:DOS
===============

120  REM  PAUL RAYMER
130  REM  3464 TOWNHOUSE DRIVE
140  REM  LAS VEGAS NV 89121

150  DIM M$(16)
160  VTAB 2: HTAB 3: PRINT ":CON:DOS - MENU OF APPLICATIONS:"
170  FOR X = 1 TO 16: READ M$(X): NEXT X
180  FOR L = 1 TO 16
190  VTAB 3 + L: HTAB 7: PRINT  CHR$ (L + 64);: HTAB 8: PRINT " = ";: HTAB
     11: PRINT M$(L)
200  NEXT L
210  PRINT : CALL  - 958: PRINT "SELECT BY LETTER (A-P) ";: GET N$:N =  ASC
     (N$) - 64: PRINT
220  INVERSE
230  VTAB N + 3: HTAB 11: PRINT M$(N)
240  NORMAL
250  VTAB 20: HTAB 1: CALL  - 958
260  VTAB 21: HTAB 1: PRINT "IS ";M$(N);" CORRECT? (Y/N) ": VTAB 22: HTAB
     4: FOR D = 1 TO  LEN (M$(N)): PRINT "-";: NEXT D: VTAB 21: HTAB 15 +
      LEN (M$(N)): GET AN$:
270  IF AN$ <  > "N" THEN 290
280  PRINT  CHR$ (7): GOTO 180
290  PRINT
300  IF N = 16 THEN  END
310  PRINT  CHR$ (4);"RUN ";M$(N)
320  END
330  DATA  LENGTH,IMPRESSIVE,CRUNCH,APPLE,SORT,APPLETREE,SHIFTY,REVERS-O,I
     NVERSORT,ZILCH,RENAMER,INVENTORY,SPEED LOCK,PERSONALIZER,REPLACER,END
      PROGRAM,SLOT MACHINE
```

manner, of course, should properly be shared with the author of CON:DOS.

Menu for CON:DOS

This program is the starting point and should be tried first. It is very likely the best program in the series!

Lines 100–150 clear the decks and decide that a maximum of 16 things will be kept is array M$. If you want to use this concept on a program with more items, or a bigger television set, you may want to change this number. Sixteen is plenty for me.

Line 160 establishes the title. If your Apple does not have 48K memory, you may wish to change copy in the print statement to "MENU". It will require changing the HTAB to 19 as well.

Line 170 reads the data statements and memorizes the stuff so you can use it time and time again without having to retype it. Also quite handy if you decide to leave out half of the programs. Just change the data statements and value from 16 down to whatever and everything will still work.

Lines 180–200 build the menu using the data stuff. The CHR$(L + 64) lets you use letters for one-key instant selection. Increasing the response of the user enables him/her to get this darn thing over with faster, so as not to miss Dallas, Dynasty or General Hospital.

Line 210 may be anticlimactic, since I spilled the beans in the previous paragraph.

Lines 220–240 provide a classy touch to the menu. The item selected is now highlighted. This kind of thing is easy when you use arrays, inverse-normal commands and an Apple. Could get real hairy if you were still trying this on your old Raaadio Shaaack Mod I.

Lines 250–260 provide the whipped cream and cherry every good program should have. When the question is asked, an underline is placed under the title selected as reinforcement of proper selection. The underline is always the correct number of dashes (hyphens, really) because of the LEN (M$(N)) thing.

Lines 270–300 determine if we should continue or start again.

Lines 310–320 run the selection.

Line 330 is the data for the program, consisting basically of the names of the programs on the disk.

I suggest that you write this program first, completely. You can save it under each of the names in the directory, if you wish, which will enable you to test it out completely. Later on, should you continue with this article, as you save each new program under its proper name, things will work out just fine.

If you wish to stop after seeing this program, I have only two words for you—INIT DISK!

Otherwise. . .

Length

This program is a very practical one. It's an original concept and certainly what the developers of COMSAT had in mind back in 1963.

LENGTH

```
100  TEXT : HOME : CLEAR
110  DIM A(21),N$(30),A$(30)
120  REM

===============
FILE NAME LENGTH
===============

130  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NV 89121


140  REM IX/XIV/MCMLXXXII

150  FOR I = 1 TO 21: READ A(I): NEXT I
160  PRINT  CHR$ (4)"CATALOG"
170  FOR X = 4 TO 21
180  FOR Y = 0 TO 29
190  IF  PEEK (A(X) + Y) = 160 AND  PEEK (A(X) + Y + 1) = 160 THEN 230
200 N$(X) = N$(X) +  CHR$ ( PEEK (A(X) + Y))
210  NEXT Y
220  IF  MID$ (N$(X),2,1) =  CHR$ (160) THEN 240
230  NEXT X
240 XX = X - 1
250  REM

=======
COUNTER
=======

260  VTAB 1: PRINT "FILE NAME LENGTH COUNTER"
270  FOR X = 1 TO XX
280  IF  LEN (N$(X)) < 8 THEN 320
290 A$(X) =  RIGHT$ (N$(X), LEN (N$(X)) - 7)
300 LE =  LEN (A$(X))
310  VTAB X: HTAB LE + 10: PRINT "(";LE;")"
320  NEXT X
330  END
340  DATA  1024,1152,1280,1408,1536,1664,1792,1920,1064,1192,1320,1448,157
     6,1704,1832,1960,1104,1232,1360,1488,1616,1744,1872,2000
```

IMPRESSIVE

```
100  TEXT : HOME : CLEAR
110  REM

==========
IMPRESSIVE
==========

120  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NEVADA 89121

130  REM  IX/XIV/MCMLXXXII

140  PRINT  CHR$ (4)"CATALOG"
150  SPEED= 100
160  FOR X = 6 TO  PEEK (37)
170 R =  INT (900 *  RND (1)) + 100
180  VTAB X: HTAB 4: PRINT R
190  FOR U = 1 TO 3:B =  PEEK ( - 16336): NEXT U
200 T = T + R
210  NEXT X
220  VTAB  PEEK (37) + 2: HTAB 9: PRINT  CHR$ (34)"TOTAL SECTORS = ";T; CHR$
     (34)
230  FOR Z = 1 TO 2000: NEXT Z: PRINT "(MUCH MORE THAN APPLE ALLOWS ON A D
     ISK)";
240  SPEED= 255
250  VTAB 3: HTAB 2: PRINT " DISK VOLUME "; INT (T / 2)
260  VTAB 2: HTAB 1: END
```

Lines 100–140 clear the program, screen and memory.

Line 150 reads (and will hopefully remember) all the data in line 340.

Line 160 displays the catalog. Notice that the CHR$(4) command is used. Lots of people would have just stuck the line (say at 101) and said D$ = CHR$(4). I don't do that because when I program, I may decide to use the letter D for something later on—like D for Delete or D for Destroy or (more likely) D for Dinner—so I just use the CHR$(4) bit. You are certainly welcome to do what you wish. After all, some of you folks may even read *Byte* magazine.

Lines 170–240 look at each line on the screen carefully. I am not too sure what happens here; this whole concept came to me one night after eating six chili dogs at Sam's Town Casino, but, if I recall, it will look at each line and remember what it saw—until it hits two blank spaces in a row (160 in PEEK talk is a space. Try subtracting 128 from 160 and see if the number is significant in regular ASCII talk). When it does, it assumes that nothing too exciting will follow and so quits looking.

Lines 250–320 print the results on the screen. Actually we are going to use the array N$( ) but print all of it *except* the first seven characters. The first person (only) who sends in the correct answer will win a quarter. Be sure to enclose your five cents entry fee to help cover my gambling expenses.

Note that we kinda cheat a bit, but it makes for easier understanding of the program. For example, line 240 decides that XX = X – 1. We could have just stuck with the X – 1, but then we would have had to change the variable in line 270. We created LE to equal LEN (A$(X)) to make line 310 easier to understand. After all we have a jillion (decimal) variables available, so why not use them?

Line 340 is the data line, which by now you have figured out contains the points on the screen which match up the VTAB and HTAB locations. Apple tells you about the TAB stuff in their book. I have a sheet of paper which shows me the memory locations of the screen. If I ever lose that piece of paper, I'm dead.

This program, if ever completed, should be saved as Length.

Impressive

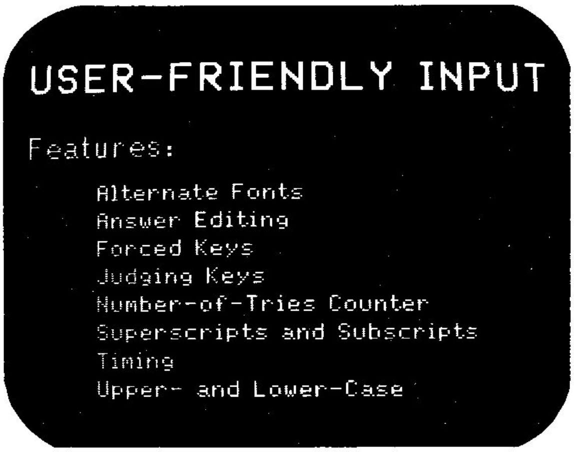
Here is the program you will want to show your friends who understand programming. Because of the highly technical nature of its concept, do not bother displaying it to computer magazine publishers, CEOs of computer companies or retail computer salespeople.

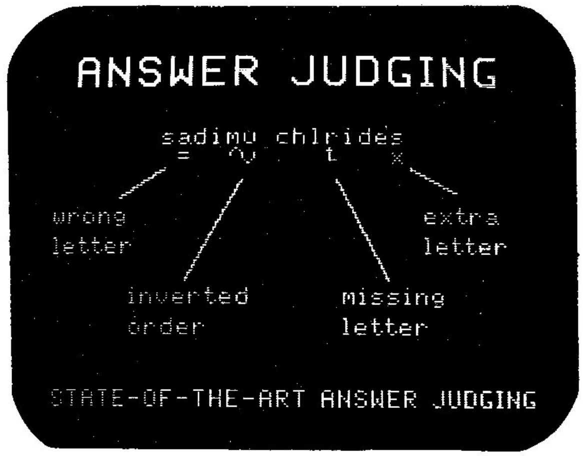
The program is very short, because—honestly—it doesn't do very much. But, what it does, it does quickly, accurately, neatly and with a flourish.

Lines 100–150 clear the decks, dis-

Now YOU can write professional quality interactive Computer-Assisted Instruction materials and simulations

EnBASIC™ can help YOU





For the Apple II plus* and //e*, with at least 48k of memory and 3.3 DOS.

Authors Paul Tenczar, Stanley Smith, and Allen Avner have produced CAI and similar user-oriented software for more than 20 years. Here are routines and authoring aids critical to preparation of high-quality, user-friendly materials.

EnBASIC adds to Applesoft* BASIC. All features of BASIC are still present.

A flexible display design allows for:

- Proportional spacing, superscripts, subscripts, underlining, and automatically backspaced diacritic marks in text
- Double or standard size characters displayed anywhere on the High Res screen
- Lower-case characters on the Apple II plus*

Advanced input handling affords you:

- State-of-the-art answer-judging
- Automatic indication of spelling and typing errors
- Character-by-character error feedback for missing, extra, or wrong letters, inverted letter order, errors in accenting, capitalization, sub- or superscripting
- Synonym lists allowed as part of expected responses

You get all these features simply by specifying a correct response and including an EnBASIC command that implements spelling checks with feedback.

The Package

A **94-page manual** containing a tutorial on use of EnBASIC with examples, implementation hints, and technical details (available separately for $20.00 which may be applied to the full purchase price of $150.00).

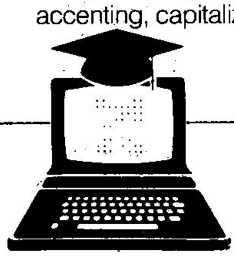
A **pocket guide** to **EnBASIC** commands

A master and back up diskettes containing: the EnBASIC augmentation program, six ready-made character sets, four sizes of English letters plus Cyrillic and Greek together with editors which allow you to design your own character sets and redefine key set functions.

A sample program diskette

*Registered trademark of the Apple Computer Company
™ A trademark of Computing Teaching Corporation

EnBASIC Package $150.00



To order or receive our catalog, call or write today!

COMPress

DEPARTMENT C
P.O. Box 102, Wentworth, NH 03282
(603) 764-5831 / 5225

For more information contact: Jane Wescott, 286 Congress Street, Boston, MA 02210 (617) 426-2224.

Circle 262 on Reader Service card.

CRUNCH

```
100  TEXT : HOME : CLEAR
110  DIM N(24)
120  REM

======
CRUNCH
======

130  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NEVADA 89121

140  REM  IX/XVIII/MCMLXXXII

150  FOR I = 1 TO 24: READ N(I): NEXT I
160  HOME
170  PRINT  CHR$ (4)"CATALOG"
180  PRINT
190  VTAB 1: HTAB 1: PRINT "ZAP OUT WHAT LETTER (A-Z)? ";: GET AN$
200  FOR X = 1024 TO 2038
210 L =  PEEK (X)
220  IF  ASC (AN$) + 128 = L THEN  POKE X,163: FOR BZ = 1 TO 7:U =  PEEK (
      - 16336): NEXT BZ
230  NEXT X
240  VTAB 1: HTAB 1: PRINT "ZAP (A-Z) <RETURN> TO CRUNCH (ESC) QUIT";: GET
     AN$: VTAB 1: HTAB 1: CALL  - 868: FLASH : PRINT "CRUNCHING": NORMAL
250  IF AN$ =  CHR$ (27) THEN 360
260  IF AN$ > "A" THEN 200
270 X = 1
280 FLAG = 0
290  FOR Y = 1 TO 38
300  IF  PEEK (N(X) + Y) = 163 THEN  POKE N(X) + Y, PEEK (N(X) + Y + 1): POKE
     N(X) + Y + 1,163:FLAG = 1: FOR BZ = 1 TO 3:U =  PEEK ( - 16336): NEXT
     BZ
310  NEXT Y
320  POKE N(X) + 39,160
330  IF FLAG = 1 THEN 280
340 X = X + 1: IF X < 24 THEN 290
350  GOTO 240
360  VTAB 1: HTAB 1: CALL  - 868: END
370  DATA  1024,1152,1280,1408,1536,1664,1792,1920,1064,1192,1320,1448,157
     6,1704,1832,1960,1104,1232,1360,1488,1616,1744,1872,2000
```

play the catalog and slow the excitement down to an acceptable pace.

Lines 160–210 perform a For... Next loop that starts at the sixth line of the catalog and goes to the last line. The PEEK (37) takes care of finding the last line of the catalog. Although it probably is defined in the Apple reference manual as imm & def/(aexpr) or some such nonsense, why not use the PEEK (37) and trust me!

Random numbers from 100 to 1000 are going to be generated at Line 170 and assigned to variable R. If you feel more comfortable, you may wish to change the value in line 170 from 900 to 899, to assure that no number greater than 999 will be created. If 1000 did come up it would be a mess. But I am a gambler and what the heck, the odds are 1000 to 1!

Those numbers are going to be placed—line for line—where the sector numbers are traditionally located. If you want to know more about sectors, write to Roger Wagner of Southwestern Data Systems, or Bert Kersey of Beagle Bros. Those programming guys in the San Diego area are really hot for sectors and can be of great help to you. They probably won't answer your letters, but if you include about three or four first-class postage stamps, they may send you a neat catalog. They each sent me one.

Line 190 makes noise. That's all it does.

Line 200 is an increment counter. See how variable names are created? I used R for "random" and now will use T for "total." You may wish to read my book, *Naming Variables: An Erotic Experience*, soon to be published by Wayne Verde of the Hencho in Mexico Publishing Company.

Line 220 prints the total sectors. No big deal—after all, it is using the T variable and has had plenty of time to do the arithmetic. The CHR$(34) is the only way I could figure out to get quotation marks on the screen. Do you have any other suggestions?

Lines 230–260 make a pithy editorial comment, re-evaluate the Disk Volume and end the program. Just in time.

This program should be saved under the name Impressive.

The Cruncher

This program is not as violent as it appears. I had originally intended to call it Apple Panic or Dueling Digits or General Ledger: An Officer and a Gentleman, but I got a very curt note from a Mr. Carlston at Broderbund Software and so this little gem is entitled The Cruncher. Has a nice ring to it, doesn't it?

One note of caution. When this program runs it does fearful things to your screen. You may wish to turn the contrast and bright controls to a more moderate position before running.

Lines 100–150 clear things and remember 24 items. These things are in the data statement line at the end of the program.

I was once told to put all data statements early in the program since it was more convenient for the Apple to find them that way and it saved something called "overhead." Nuts! After spending a couple of thousand bucks on my computer, let *it* do something for *me*! Besides, I think if I stick it at the end of the program—and get it out of the way—I don't have to bother with it, and the Apple can find it easily no matter where it is.

Lines 160–180 clear the screen and display the catalog.

Line 190 positions the cursor at the top of the screen and states the name of the game.

Lines 200–230 are the heart of the program. Your computer is going to beat its brains out looking at every screen location from the upper left corner to the lower right. (That's the 1024 to 2038.) If it finds the letter you indicated as AN$, the fur will fly! First of all good ole Apple adds 128 to the ASCII value of your AN$. Peeks need 128 more to work than regular ASCII things do.

Now, if it finds one, it will replace it with a #. I am not sure if this is called a number sign or a tic tac toe sign, but for sure it is ASC (163). A noise is made to let you know something is happening—and the search continues.

When all this feverish excitement ends, the program gracefully goes to...

Line 240, which presents a submenu. You will have the choice for zapping more letters, starting to crunch them or quitting. If you want to zap some more, be my guest.

The line assumes you will be crunching, so all the instructions are contained for crunch in that line.

Lines 250–260 bring the program back to reality. If Quit command is issued, the program will end. If any letter of the alphabet (greater than A),

Emulates these terminals exactly.

IBM 3101
DEC VT100, VT52
Data General D200
ADDS Regent 20, 25, 40
Hazeltine 1400, 1410, 1500
Lear Siegler ADM-3A, ADM-5
TeleVideo 910
Teletype Model 33 KSR

Apple is a trademark of Apple Computer, Inc.



New File Transfer Language

BREAK
CATALOG
CHAIN
CONFIGURE
CONNECT
CONVERSE
DIAL
END
HANGUP
LOG
MONITOR
NOLOG
ONERR
PAUSE
PROMPT
RECEIVE
REMARK
RETRIES
SEND
SPECIAL
SPEED
TIMEOUT
XMIT:WAIT

Supports these interface boards.

Apple Communications Card
Apple Parallel Printer
Apple Serial Interface
Apple Super Serial Card
Bit 3 Dual-Comm Plus™
CCS 7710, 7720, 7728
Hayes Micromodem II™
Hayes Smartmodem™ 300, & 1200
Intra Computer PS10
Mountain Computer CPS Card™
Novation Apple-Cat II™ 300 & 1200
Orange Micro Grappler™
Prometheus VERSAcard™
SSM ASIO, APIO, AIO, AIO II™

Supports your 80-column hardware.

ALS Smarterm™
Bit 3 Full-View 80™
Computer Stop Omnivision™
M&R Sup'R'Terminal™
STB Systems STB-80™
Videx Videoterm™
Vista Computer Vision 80™
Wesper Micro Wizard 80™

Your host computer won't know the difference!

Softerm provides an *exact* terminal emulation for a wide range of CRT terminals which interface to a variety of host computer systems. Special function keys, sophisticated editing features, even local printer capabilities of the terminals emulated by Softerm are fully supported. Softerm operates with even the most discriminating host computer applications including video editors. And at speeds up to 9600 baud using either a direct connection or any standard modem.

Unmatched file transfer capability

Softerm offers file transfer methods flexible enough to match any host computer requirement. These include *character protocol* with user-definable terminator and acknowledge strings, block size, and character echo wait, and the intelligent *Softrans*™ protocol which provides reliable error-free transmission and reception of data. The character protocol provides maximum flexibility for text file transfers. *Any* type file may be transferred using the Softrans protocol which provides automatic binary encoding and decoding, block checking with error recovery, and data compression to enhance line utilization. A FORTRAN 77 source program is supplied with Softerm which is easily adaptable to any host computer to allow communications with Softerm using the Softrans protocol.

Softerm file transfer utilizes an easy to use *command language* which allows simple definition of even complex multiple-file transfers with handshaking. Twenty-three high-level commands include *DIAL, CATALOG, SEND, RECEIVE, ONERR, HANGUP, MONITOR* and others which may be executed in immediate command mode interactively or from a file transfer macro command file which has been previously entered and saved on disk.

Built-in utilities

Softerm disk utilities allow DOS commands such as CATALOG, INIT, RENAME, and DELETE to be executed allowing convenient file maintenance. Local file transfers allow files to be displayed, printed, or even copied to another file without exiting the Softerm program. Numerous editing options such as tab expansion and space compression are provided to allow easy reformatting of data to accommodate the variations in data formats used by host computers. Softerm supports automatic dialing in both terminal and file transfer modes. Dial utilities allow a *phone book* of frequently used numbers to be defined which are accessed by a user-assigned name and specify the serial interface parameters to be used.

Online Update Service

The Softronics Online Update Service is provided as an additional support service at no additional cost to Softerm users. Its purpose is to allow fast turnaround of Softerm program fixes for user-reported problems using the *automatic patch facility* included in Softerm as well as a convenient distribution method for additional terminal emulations and I/O drivers which become available. *User correspondence* can be electronically mailed to Softronics, and *user-contributed* keyboard macros, file transfer macros, and host adaptations of the Softrans FORTRAN 77 program are available on-line.

Most advanced communications software available

Just check Softerm's 300 page user manual. You simply can't buy a more sophisticated package or one that's easier to use. Available now for only $150 from your local dealer or Softronics, Inc.

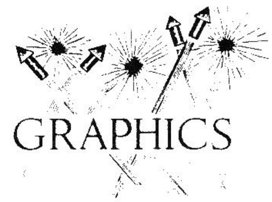
SOFTRONICS

6626 Prince Edward, Memphis, TN 38119, 901-755-5006

Circle 266 on Reader Service card.

The Routine Machine™





A Versatile Programming Utility for the Apple II, II+, IIe.

PRINT USING





Now, from the programming experts at S.D.S., an easy-to-use way of putting the POWER and SPEED of machine language routines in YOUR OWN APPLESOFT PROGRAMS!

ROUTINE MACHINE does all the work for you — no knowledge of machine language programming, whatsoever, is required. Simply choose the routine you need from an ever-growing library, and Routine Machine will effortlessly put them in your program, and all done transparently! No need to be aware of or bother with BLOAD's, HIMEM:, etc.

Best of all, with just this starter package, you'll have the routines to put High Resolution **graphics** and **sound** in your programs immediately! Also included is a versatile **print using** module to banish the "decimal point demons" forever! To round out the package, we've also included powerful **search** and **sort** routines (for single dimension arrays; Search: 1000 elements in 1 second, Sort: 1000 elements in 90 seconds), and a number of other often-needed routines as well (30 routines in all).

OUR GUARANTEE

IF YOU DON'T SAVE MORE THAN THE PURCHASE PRICE OF 'ROUTINE MACHINE' IN YOUR OWN PROGRAMMING TIME IN THE FIRST 30 DAYS YOU OWN IT, SIMPLY RETURN IT FOR A COMPLETE REFUND, NO QUESTIONS ASKED!

Some of the other routines in The Routine Machine (plus others not listed) are:

SWAP: Swaps two string or numeric values.

TEXT OUTPUT: Prints with no "word break" on screen.

STRING INPUT: Input any string, regardless of commas, etc.

ERR: Stack fix for Applesoft ONERR handling.

GOTO, GOSUB: Allows computed statements. Example: **GOTO X✳5** or **GOSUB X✳5**.

BLOAD: Load any binary file 5 times faster than normal. Hi-Res pictures load in under 2 seconds.

RESET HANDLER: Treats RESET with ONERR; or will RUN or reboot disk.

HI-RES ASCII: Character set for mixing text Hi-Res graphics.

TURTLE GRAPHICS: Versatile Hi-Res graphics routines for easy drawing of Hi-Res figures.

SYSTEM REQUIREMENTS: 48K Apple II, DOS 3.3 w/FP ROM or language card.

PRICE: $64.95 California residents add 6% sales tax plus $2.50 shipping.

Send for free information on additional Subroutine Library Disks.

Apple II and Applesoft are both trademarks of Apple Computer, Inc.

southwestern data systems™

P.O. BOX 582-I • SANTEE, CALIFORNIA 92071 • TELEPHONE: 619/562-3221

Circle 249 on Reader Service card.

the program will GO TO 360 and do its thing again. Otherwise, the crunching starts.

Lines 270–340 provide the thrill and adventure to an otherwise erudite program. This program will, if I haven't goofed, now look for all the tic tac toe signs and replace each one with the character *next to it*, on its right. Each time it finds a # it gets so excited it makes a noise. It keeps on doing this until all the #s are gone.

The use of the variable FLAG (or FL if you are an old hacker) insures that return is made to each line until *all* of the #s are gone. Without this little gimmick, only one letter would be zapped in each line—and the program would take forever. And it's long enough as it is.

Line 350 returns you to Control Central.

Line 360 wraps things up neatly and ends the program with the cursor on the top line.

Line 370 is data. The screen locations needed to find things.

This program should be saved under the name Crunch.

Apple Maker

This simple program is easy to type, easy to understand, easy to think of. You may wonder why I would even bother.

I have had a terrible time trying to get in contact with the Apple Computer company. (I wonder if there *is* an Apple company sometimes. Do you? Or have the two Steves figured out how to have the Apples replicate themselves using THAT™ program, so that no people are required to answer letters?

This program may help show them I am not angry—merely frustrated. And you nice folks out there will benefit from it by having a fun thing to do until the reports come back on your medical exams.

Lines 100–130 clear the screen and display the catalog. The Line 120 is a bit jazzy and totally uncalled for. A simple REM PAUL RAYMER should have been sufficient.

Lines 140–190 are a For. . .Next loop from 1 to 4 less than the total number of entries on the catalog.

Lines 150–160 create my favorite routine set. It allows proper lineup of single and double digit numbers, to prevent the single digit numbers from being too far to the left. You'll see. Each time a proper number is printed a bell rings. Don't forget the semi-colon at the end of the bell!

A slight delay permits the grandeur of the moment to be appreciated.

Line 200 picks the line where most folks have VOLUME 254 on their disks. I have COMPUTER EXPERT AT WORK on mine. Makes my computer tax-deductible, my garage mechanic told me. Then it prints two characters from left to right. These characters are based on my Nevada computer, which may be different than yours. I will be anxious to find out what displays you will get.

Line 210 simply erases the old stuff.

Line 220 places the second character into position.

Lines 230–270 wrap up the loose ends.

Save this program under the title Apple.

APPLE MAKER

```
100  TEXT : HOME : CLEAR
110  REM

===========
APPLE MAKER
===========

120  REM

 P
RAYMER
 U
 LAS VEGAS


130  PRINT  CHR$ (4)"CATALOG"
140  FOR X = 1 TO  PEEK (37) - 4
150  IF X < 10 THEN  VTAB 4 + X: HTAB 4: CALL  - 868: PRINT "OO";X;" APPLE
     ": GOTO 170
160  VTAB 4 + X: HTAB 4: CALL  - 868: PRINT "O";X;" APPLE"
170  PRINT  CHR$ (7);
180  FOR Z = 1 TO 500: NEXT Z
190  NEXT X
200  FOR Y = 2 TO 18: VTAB 3: HTAB Y: PRINT  CHR$ (93) CHR$ (91)
210  VTAB 3: HTAB Y - 1: PRINT  CHR$ (32): FOR Z = 1 TO 100: NEXT Z: NEXT
     Y
220  VTAB X + 4: HTAB 3: PRINT  CHR$ (91)
230  VTAB X + 3: HTAB 1: END
240  REM
250  REM  PAUL RAYMER
260  REM  3464 TOWNHOUSE DRIVE
270  REM  LAS VEGAS NV 89121
```

Sort

This is a very short program. It is not typical of the great detail, intricate design and careful programming other sections in CON:DOS employ. It has been inserted primarily because I had a bad day when I wrote this, and—like so many authors—I try to put a bit of real life, of realism, of "now," into each program I write. I certainly hope you don't!

Lines 100–130 clear program.

Line 140 decides how many times whatever it is will be done, will be done.

Line 150 is the main algorithm of the program. This uses the same high degree of accuracy and reliability employed by the TV ratings companies to insure that the networks continue to provide the level of entertainment we now get—and deserve. Note the clever

SORT

```
100  TEXT : HOME : CLEAR
110  REM

====
SORT
====

120  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NV 89121

130  REM  IX/XIV/MCMLXXXII

140  FOR X = 1 TO 18
150  VTAB X: HTAB X: PRINT "SORRY";: FOR A = 1 TO X: PRINT ".";: NEXT A: PRINT
      CHR$ (7)
160  NEXT X
170  PRINT : PRINT "OUT OF SORTS...TODAY"
```

use of a nested loop and how it adroitly and dexterously handles the period without being gauche or sinister in any manner.

Line 160 completes the loop.

Line 170 editorializes. Program ends.

Program should be saved as Sort.

Catalog Alphabetizer

You have seen many programs advertised which will alphabetize your catalog. Most of them are good, a few excellent, and some definitely poor. There really isn't much of a market for this type of program, particularly with good programmers, who are able to think of titles in alphabetical order while they are programming. It can pose a problem to the new programmer, however, and so in a moment of graciousness this fascinating program is presented.

Lines 100–180 clear screen and put 16 items into memory from the data statement at line 430. The N$(30) is just in case you want an array of 30 characters later on—you'll be ready.

CATALOG ALPHABETIZER

```
100  TEXT : HOME : CLEAR
110  DIM A(16),N$(30)
120  REM

===================
CATALOG ALPHABETIZER
===================

130  REM

 L        V
PAUL  RAYMER
 S        G
    NEVADA
        S

140  REM  3464 TOWNHOUSE DRIVE
150  REM  LAS VEGAS NV 89121

160  REM

IX/XIIII/MCMLXXXII

170  FOR I = 1 TO 16: READ A(I): NEXT I
180  PRINT  CHR$ (4)"CATALOG"
190 R =  INT (3 *  RND (1)) + 1
200  IF R = 1 THEN R$ = " HANG IN THERE "
210  IF R = 2 THEN R$ = " PLEASE STAND BY "
220  IF R = 3 THEN R$ = " ONE MOMENT PLEASE "
230  VTAB  PEEK (37) + 2: HTAB 5: FLASH : PRINT R$: NORMAL
240  VTAB 3: HTAB 1: PRINT "ALPHABETIZING CATALOG..."
250  FOR X = 1 TO 16
260  FOR Y = 0 TO 29
270 N$(X) = N$(X) +  CHR$ ( PEEK (A(X) + Y))
280  NEXT Y
290  IF  MID$ (N$(X),2,1) =  CHR$ (160) THEN 310
300  NEXT X
310 XX = X - 1
320  REM

====
SORT
====

330 S = 1
340  IF  RIGHT$ (N$(S),23) >  RIGHT$ (N$(S + 1),23) THEN X$ = N$(S + 1):N$
     (S + 1) = N$(S):N$(S) = X$: GOTO 330
350 S = S + 1: IF S = XX THEN 370
360  GOTO 340
370  VTAB 5: HTAB 1: CALL  - 958
380  FOR X = 1 TO XX
390  VTAB X + 4: HTAB 1: PRINT N$(X)
400  NEXT X
410  VTAB  PEEK (37) + 1: CALL  - 958: PRINT : PRINT "DONE "; CHR$ (7)
420  END
430  DATA  1536,1664,1792,1920,1064,1192,1320,1448,1576,1704,1832,1960,110
     4,1232,1360,1488,1616,1744,1872
```

Line 130 is gingerbread, and the author's name and address would certainly have been more than enough. Remarks can't be entered that way anyhow.

Line 190 establishes a random number from 1 to 3.

Lines 200–220 assign copy lines based on the random selection made.

This program—although not long in context or in reason—takes some bit of time in operation. The use of the R$ is an amusing bit of fluff to help divert you from the boring wait during the bubble sort.

Line 230 compounds the felony by printing a flashing word. Looks like something exciting is happening.

Line 240 finally tells us what this thing is all about.

Lines 250–310 read the screen, one line at a time, and collect the names. (That's where the 30 comes in from line 110.) The double-zero gimmick is used again. If CHR$(160) shows up twice in a row, you can assume no more words follow. You track it down this time in another manner, by using a MID$ (N$(X),2,1) approach.

Lines 330–370 perform the sort. The N$( ) array is sorted according to the good old standby bubble sort. There are many methods faster than this, but really—what's the hurry? I don't know about you folks, but there isn't much to do here in Las Vegas after dark, except sit at a computer and thrill at the action on a 9-inch black and white monitor.

Lines 380–420 print out the glorious results. This looks better than it sounds, I think. To the old hands, it may appear rather primitive, but to new kids on the block at least something is happening without taking a lifetime to enter into the keyboard. Besides, anyone who writes programs

SHIFTY

```
100  TEXT : HOME : CLEAR
110  REM

======
SHIFTY
======

120  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NEVADA 89121

130  REM  IX/XVII/MCMLXXXII

140  PRINT  CHR$ (4)"CATALOG"
150  VTAB 3: HTAB 1: PRINT "CATALOG SHIFTER"
160  VTAB 1: HTAB 1: INPUT "SHIFT HOW MANY SPACES? (1-10) ";S
170  VTAB 1: HTAB 1: CALL  - 868
180  FOR X = (2038 - S) TO 1024 STEP  - 1
190 P =  PEEK (X): POKE X + S,P
200  NEXT X
210  FOR X = 1 TO 23: VTAB X: HTAB 1: PRINT  SPC( S): NEXT X
220  VTAB 3: HTAB 1 + S: CALL  - 868: PRINT "CATALOG SHIFTED"
230  VTAB 23: HTAB 1: END
```

is more likely to be at the keyboard right now than reading this article! I know I am.

Line 430 is the list of screen locations we use to get things lined up properly. We put all of them in one long line of data since, if we put each item on a line by itself, we would have to enter the line as DATUM 1536 (et cetera), and that ain't going to work nohow.

This program should be saved under the title Appletree.

Shifty

I was working at my computer and noticed that the relationship of the monitor and my line of sight created an uncomfortable angle. I wrote this program to correct that problem.

While the total development time was something like 144 hours (which may seem gross to many of you in the mercantile industries), I felt that it was worth it.

I later discovered that I could have moved the TV set a little to the left and accomplished the same results.

Lines 100–150 clear screen and display the catalog.

Line 160 inquires as to how many spaces the catalog will be shifted. The response is cleverly retained by variable N. This is clearly explained in the Apple reference manual as INPUT (string ;) var ((, var)). OK?

REVERSE CATALOG

```
100  TEXT : HOME : CLEAR
110  DIM A(16),N$(30)
120  REM

===============
REVERSE CATALOG
===============

130  REM

     PAUL RAYMER

     PAUL'S ELECTRIC COMPUTER
     3464 TOWNHOUSE DRIVE
     LAS VEGAS NEVADA 89121

140  REM  IX/XIII/MCMLXXXII

150  FOR I = 1 TO 16: READ A(I): NEXT I
160  PRINT  CHR$ (4)"CATALOG"
170  FOR X = 1 TO 16
180  FOR Y = 0 TO 29
190 N$(X) = N$(X) +  CHR$ ( PEEK (A(X) + Y))
200  NEXT Y
210  IF  MID$ (N$(X),2,1) =  CHR$ (160) THEN 230
220  NEXT X
230 XX = X - 1
250  VTAB 4: HTAB 1: CALL  - 958
300  FOR X = 1 TO XX
310  FOR B = 1 TO 30
320  VTAB X + 4: HTAB 40 - B: PRINT  MID$ (N$(X),B,1)
330 U =  PEEK ( - 16336): NEXT B
340  NEXT X
350  FOR A = 1280 TO 1319: POKE A,32: NEXT A
380  FOR Z = 1 TO 500:U =  PEEK ( - 16336): NEXT Z
390  VTAB 3: HTAB 18: PRINT "DONE";: VTAB 23: HTAB 1
400  END
410  DATA  1536,1664,1792,1920,1064,1192,1320,1448,1576,1704,1832,1960,110
     4,1232,1360,1488,1616,1744,1872
```

Line 170 just cleans up that line with our old college chum CALL −868.

Lines 180–200 now scan backwards across the video screen and shift each letter the indicated number to the right. No big disasters can happen since very seldom are any file titles more than 20 characters long. If they are, watch out!

Lines 210–230 clean up the debris caused by the cavalier hacking around. Things look nice and neat as the program ends.

This program should be saved as Shifty.

Reverse Catalog

Those of us who may be left-handed, or left-footed, or left-eyed, are certainly left-out of many computer programs.

Ever notice that all paragraphs start at the left of the screen and go to the *right*, or that the line numbers are on the left and the good stuff is at the *right*? This program—in some small (or perhaps microscopic) manner may help make up for that.

Lines 100–160 clear the screen and read the data into memory (from left to right, of course).

Lines 170–220 read the lines of the catalog and put the file names into array N$( ).

Lines 230–340 print out the array, but each line is printed in reverse. This all takes place in line 320.

Line 330 has some important function, but I just spilled a cup of coffee on the printout and can't read it. Comments from you readers would be appreciated. Please keep comments in a friendly tone, if possible.

Line 350 is really neat! This is about as close as I can come to machine language. Or care to. There is another, more traditional, way to do this, but this can open up whole new avenues of adventure. If not avenues, perhaps streets—or alleys.

Line 380 also has coffee stains. Please refer to line 330.

Lines 390–400 complete the program just like the IBM, CP/M based machines and other real computers do.

Line 410 deals with data again.

This program should be saved as Revers-O.

Inversort

It's almost a shame to take the money for writing this one.

What we have here is what is known in the TV biz as a spin-off or in literary circles as a sequel.

In my case this is known as a laziness.

This is almost like one of the other programs (described in laborious detail earlier) which sorts the catalog. Only it does it in inverse.

There are a few discreet changes.

Line 180 is a diversionary tactic to make you think something is happening. It ain't.

Except for the *inverse, normal* stuff it is pretty much the same as the Applesort program.

That's what's so nice about all of this. With just a little ingenuity, creativity and cheating you can make new programs out of old ones.

Save this program under the title Inversort.

INVERSORT

```
100  TEXT : HOME : CLEAR
110  DIM A(16),N$(30)
120  REM

=========
INVERSORT
=========

130  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NEVADA 89121

140  REM  IX/XVII/MCMLXXXII

150  FOR I = 1 TO 16: READ A(I): NEXT I
160  PRINT  CHR$ (4)"CATALOG"
170  INVERSE : VTAB 1: HTAB 1: PRINT  SPC( 39)
180  FLASH : FOR P = 1 TO 7:R =  INT (39 *  RND (1)) + 1: VTAB 1: HTAB R: PRINT
     "*";: NEXT P: NORMAL
190  FOR X = 1 TO 16
200  FOR Y = 0 TO 29
210 N$(X) = N$(X) +  CHR$ ( PEEK (A(X) + Y))
220  NEXT Y
230  IF  MID$ (N$(X),2,1) =  CHR$ (160) THEN 250
240  NEXT X
250 XX = X - 1
260  REM

====
SORT
====

270 S = 1
280  IF  RIGHT$ (N$(S),23) >  RIGHT$ (N$(S + 1),23) THEN X$ = N$(S + 1):N$
     (S + 1) = N$(S):N$(S) = X$: GOTO 270
290 S = S + 1: IF S = XX THEN 310
300  GOTO 280
310  VTAB 5: HTAB 1: CALL  - 958
320  FOR X = 1 TO XX
330  FOR B = 1 TO 30
340  VTAB X + 4: HTAB B: IF  MID$ (N$(X),B,1) <  > CHR$ (160) THEN  INVERSE

350  PRINT  MID$ (N$(X),B,1): NORMAL
360  NEXT B
370  NEXT X
380  FOR Z = 1 TO 2000: NEXT Z
390  VTAB  PEEK (37) + 2: FLASH : PRINT " DONE ": NORMAL
400  VTAB 1: HTAB 1: CALL  - 868
410  VTAB 23: HTAB 1: END
420  DATA  1536,1664,1792,1920,1064,1192,1320,1448,1576,1704,1832,1960,110
     4,1232,1360,1488,1616,1744,1872
```

Zilch

Every group of computer programs should have one that is fun and novel. This may not be the one in the CON:DOS series.

Three things may be learned after you finish programming Zilch.

1. You'll see how graphics can be depicted with regular text characters. I call this Text-Graphics.

2. You'll discover how to put secrets into your programs so that the casual peruser of listings will not readily decipher your codes. Great for adventure games you may write instead of buying them from Scott Adams.

3. Don't believe everything you read. There is no third reason.

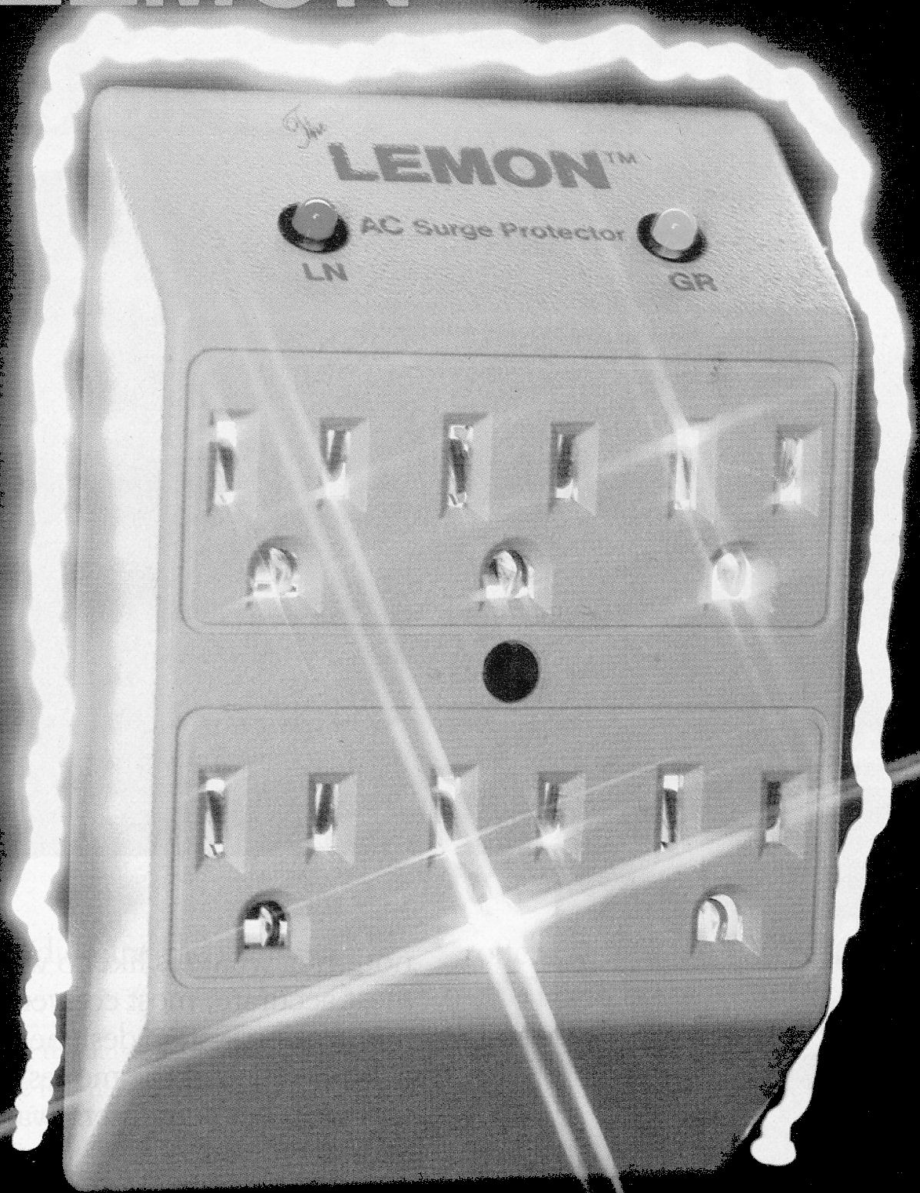
Lines 100–140 clear screen and establish a value for E$. Mode goes to *inverse*.

Lines 150–170 make a box around the perimeter, then the program waits. This is the place some crude person might suggest making "bigger boxes." Since this is a family publication, no such remark will be made here. At this time.

Line 180 is a noise loop carried beyond the limits of good taste.

Lines 190–230 establish context of

THE LEMON™ SOURS SURGES



Our crop-The Lemon™, The Lime™, and The Orange™ are designed to eliminate undetected submicrosecond overvoltage transients from electrical circuits. Commonly referred to as "spikes", or "glitches", these transients can cause hardware and software damage to unprotected circuits.

Today's electronic products are often microprocessor controlled - mini and micro computers, televisions, video cassette recorders - to name a few. Each of these products is sensitive to fluctuations in electrical power lines. Power switching devices such as refrigerators coming on and off or air conditioners starting up can be responsible for a momentary surge or spike of electricity in a circuit. Even your local utility stepping-up transformers to add power at peak load times or an electrical storm passing through can trigger surges. Such surges can cause equipment to falter at times, not to work at peak performance or fail completely. An entire data base can be lost.

Now you can prevent this from happening to you with an AC Surge Protector from Electronic Protection Devices. Each Protector is a solid state clamping device with 6 outlets utilizing modern high speed semiconductor technology. Using our Protectors is as simple as plugging it into any standard three wire duplex outlet then plugging what needs protection into it. Each Protector exceeds the IEEE 587-1980 Guide for Surge Voltages in Low Voltage AC Power Circuits.

When you compare the cost of computer hardware, software and your time with the price of a Protector (from $59.95 to $139.95), you'll want to sour your surges with one of the AC Surge Protectors from EPD, which are available through your local dealer.

Electronic Protection Devices
5 Central Avenue
Waltham, Massachusetts 02154

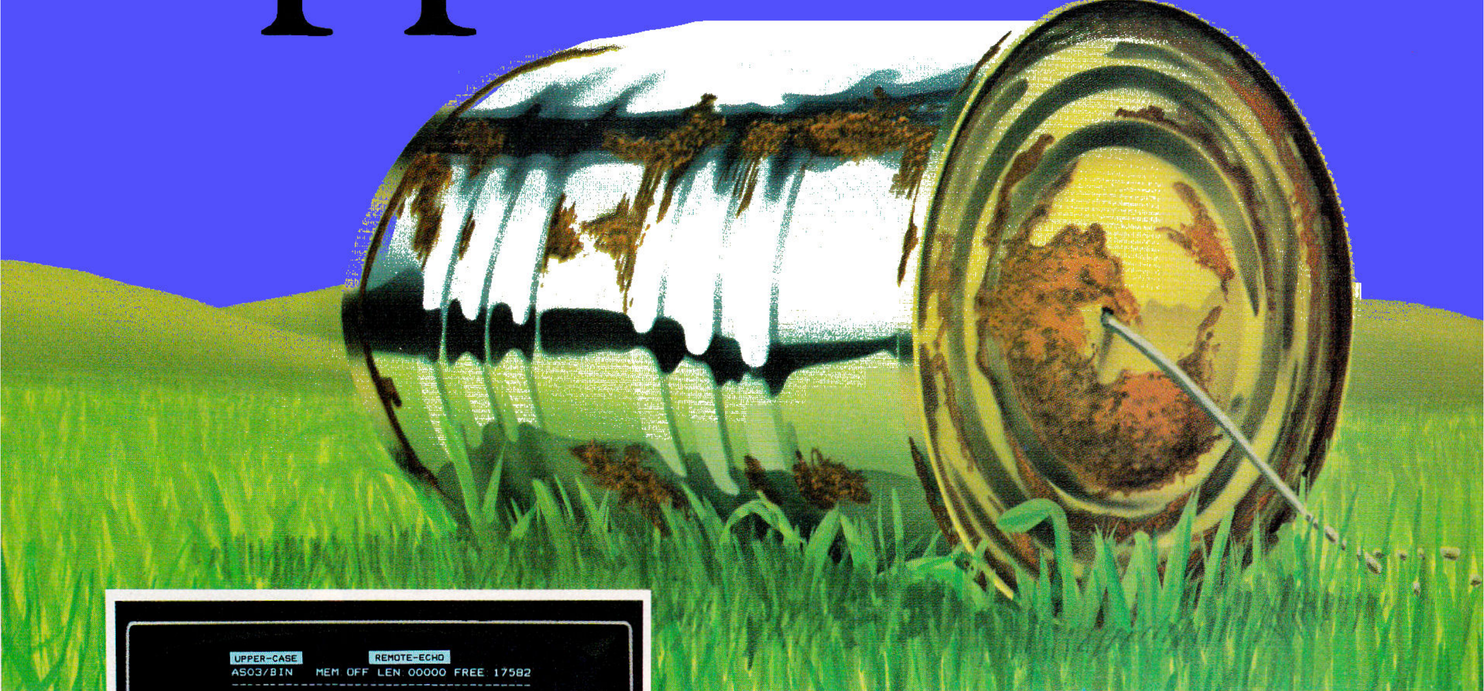
In Massachusetts Call:
(617) 891-6602

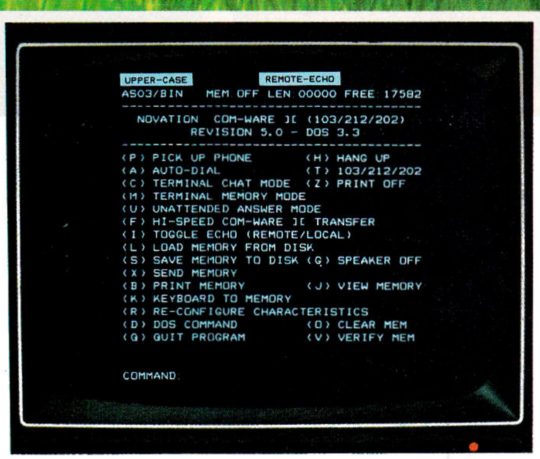
Outside Massachusetts Call:
1-800-343-1813

Dealer Inquiries Invited

Circle 25 on Reader Service card.

Apple®-Cat™ II





The Cat system lets you slip something into your Apple II no other modem offers—a complete range of speeds from 110, 300 and 202 half-duplex—to full duplex 212 .

Either way, you have state-of-the-art LSI technology. And it means you can start right off with the most advanced system available. Or you can trim your investment, yet always have the option to move up at any step with absolutely no compromises in quality.

Com-Ware™ software is part of the package.

Five minutes after you've booted up the Novation Com-Ware you'll have a good notion of what it's like to work with the best, most accurate, most convenient personal communication system designed for your Apple.

It makes all of the moves you need to work with another computer, swap programs, access data, whatever.

And it's simple to operate. Just follow the menu. No programming, no fussing. It's all there.

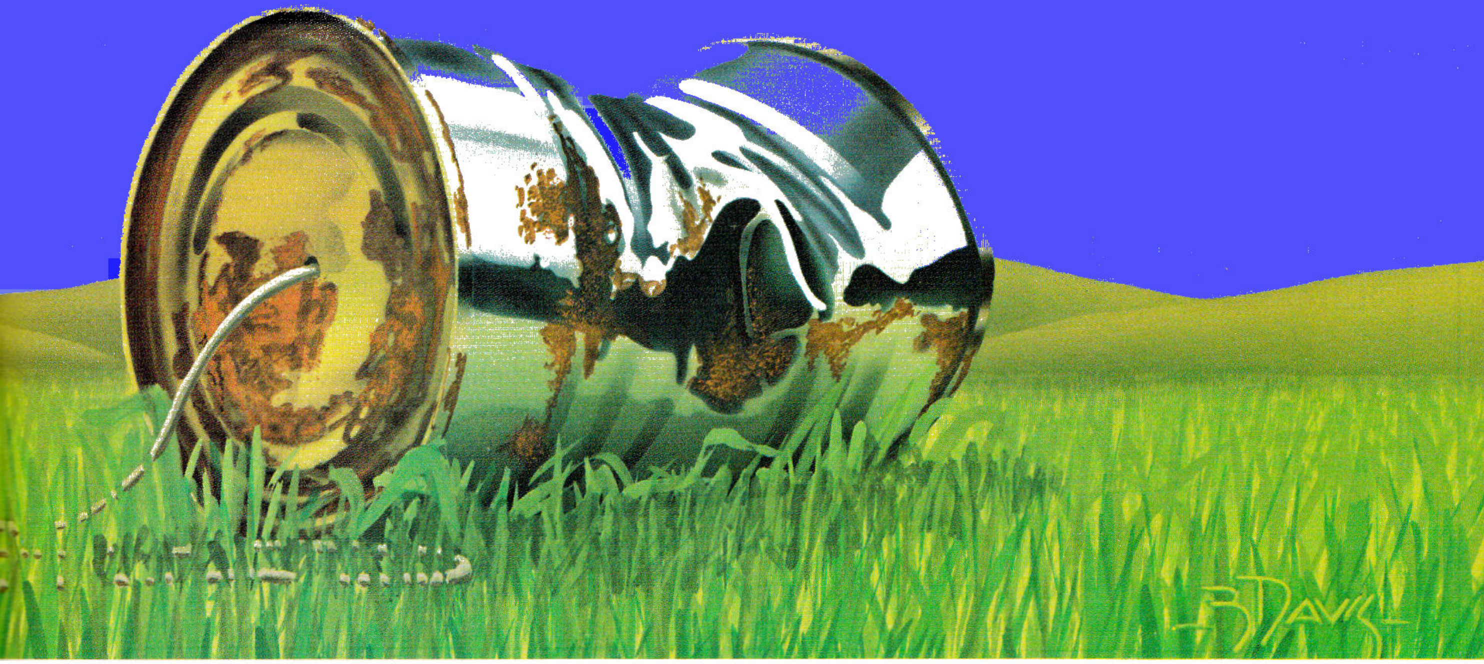
New—telephone directory.

Our engineers have done it again. They've expanded our Com-Ware. Now included: a time-saving directory of 26 telephone numbers with terminal configurations all selected and stored for auto dialing. Handy.

And some nice extras. Print-out during communication, a non-destructive memory mode, a changeable "welcome" message for automatic answer. There's more.

If you have an Apple-Cat II and our earlier Com-Ware (4.4 or earlier), you should really have the new one (5.0). As always, it's free.

It's the modem/communication system you grow into, not out of.



Just send us your old diskette and we'll send you the new one.

It can make all the moves you want.

Start with 110, 300, 202 half-duplex. *Or*—add 212 full duplex and move data four times faster with accurate, block by block verification. / Automatically send and receive anytime—including the middle of the night when line charges are lowest. / Use 80 or 40 column format. / Set for local or remote echo. And more.

Two more reasons to move right now.

Packed with every Apple-Cat II is a list of options. You get your choice of any one—and save up to $40.

Also, if you've ever wanted to take a look at **The Source,**[SM] this is your chance. With your Apple-Cat II you get a sampler subscription offer. It's limited, but it gives you a taste.

They're at your dealer.

He has all of the details. See him now.

Easy installation.
In less than 10 minutes, you're talking to the world.



More features than any other modem.

• Full range of communication baud rates—up to 1200 (Bell System 100, 202 or 212 series compatible) • Full or half duplex operation • Complete Com-Ware system on a single diskette. Also, wide assortment of excellent software available from other sources • All automatic functions—auto dial (pulse or Touch Tone), redial, auto answer and disconnect • It's a telephone with speaker monitor—switch between data and voice. For regular use, it's a handy intelligent phone with auto-dial • Touch Tone receiver • Built-in BSR X-10 Controller • Remote control for external cassette tape recorder • Works with other Apple parallel or serial printer interface cards • Constant status display on screen • Binary or text modes • Single card installation for Apple Cat II and an additional card for the 212 upgrade • FCC certified built-in phone line interface (PLI) Module • New full Duplex 212 option.

Novation

Novation, Inc.
18664 Oxnard Street
Tarzana, CA 91356

(800) 423-5419 • In California: (213) 996-5060

Circle 272 on Reader Service card.

M$ and then strip search the string using the MID$ function, an action now being heard by the Supreme Court to determine if it violates anyone's rights.

Line 240 displays the author's name, neatly centered.

This program, if saved at all, should be saved as Zilch.

ZILCH

```
100  TEXT : HOME : CLEAR
110  REM
120  REM  ZILCH
130  REM
140 E$ =  CHR$ (32): INVERSE
150  FOR X = 1 TO 40: VTAB 1: HTAB X: PRINT E$: VTAB 23: HTAB X: PRINT E$;
     : NEXT X
160  FOR X = 1 TO 23: VTAB X: HTAB 1: PRINT E$: VTAB X: HTAB 40: PRINT E$;
     : NEXT X: NORMAL
170  FOR Z = 1 TO 1000: NEXT Z
180  FOR BZ = 1 TO 50:U =  PEEK ( - 16336): FOR Z = 1 TO 100: NEXT Z: NEXT
     BZ
190 M$ = ".DEDNE MARGORP  .GNIHTON SNAEM HCLIZ"
200  FOR X =  LEN (M$) TO 1 STEP  - 1
210  VTAB 22: HTAB 39 - X: PRINT  MID$ (M$,X,1)
220  FOR Z = 1 TO 50: NEXT Z
230  NEXT X
240  REM

================
   PAUL RAYMER
3464 TOWNHOUSE DR
   LAS VEGAS NV
      89121
================
```

Catalog File Renamer

I picked up a book at the library and started to read about proportional plus integral controllers and flexed-reference modulation when I suddenly realized I had taken someone else's book home by mistake.

Exchanging the book for the *Smurf's Adventure in Computerland*, I was much more at home and greatly inspired, so I decided to design (write) another program.

After the excitement of the little Blue people running around Computerland (the storybook place, not the store), and thinking about great novelists like Dr. Seuss and the creators of that little green frog who loves my idol, Miss Piggy, I realized that what was missing in the average catalog listing of file names was *excitement*.

Why must the catalog disk look so dull?

Take just a moment and boot up something you have. What will you see? Here is a quick look at the catalog of a softdisk I've pulled out at random:

```
*A MAIN.TITLE
*A A.SHORT.ADVENTURE
*A HIST-R-XTME
*B OBJ.UR.HNR
B* ERR
*I HELLO
*I COME HERE OFTEN
*A PENTAGON FILES A-L
```

Dull, right?

That's why the File Renamer was created. To add a little spice to your otherwise dull files.

RENAMER

```
100  TEXT : HOME : CLEAR
110  DIM A(24),N$(30),MI$(30),A$(30)
120  REM

====================
CATALOG FILE RENAMER
====================

130  REM

PAUL RAYMER
3464 TOWNHOUSE DRIVE
LAS VEGAS NV 89121



140  REM

IX/XIIII/MCMLXXXII

150  FOR I = 1 TO 24: READ A(I): NEXT I
160  PRINT  CHR$ (4)"CATALOG"
170 L =  PEEK (37)
180  VTAB 3: HTAB 1: CALL  - 868: PRINT "------>AUTO FILE RENAMER"
190  FOR X = 1 TO L
200  FOR Y = 0 TO 29
210  IF  PEEK (A(X) + Y) = 160 AND  PEEK (A(X) + Y + 1) = 160 THEN 250
220 N$(X) = N$(X) +  CHR$ ( PEEK (A(X) + Y))
230  NEXT Y
240  IF  MID$ (N$(X),2,1) =  CHR$ (160) THEN 260
250  NEXT X
260 XX = X - 1
270  REM

=======
RENAMER
=======

280  FOR X = 1 TO XX
290  IF  LEN (N$(X)) < 8 THEN 310
300 A$(X) =  RIGHT$ (N$(X), LEN (N$(X)) - 7)
310 LE =  LEN (A$(X))
320  FOR RE = 1 TO LE
330 MI$(RE) =  MID$ (A$(X),RE,1)
340  NEXT RE
350  FOR P = 1 TO LE
360 RA =  INT (LE * RND (1)) + 1
370  IF MI$(RA) = "JACKPOT" THEN 360
380  VTAB X: HTAB P + 7: PRINT MI$(RA):MI$(RA) = "JACKPOT"
390  NEXT P
400  NEXT X
410  END
420  DATA 1024,1152,1280,1408,  1536,1664,1792,1920,1064,1192,1320,1448,15
     76,1704,1832,1960,1104,1232,1360,1488,1616,1744,1872,2000
```

Lines 100–150 clear the screen and stuff and load the screen location numbers into memory.

Lines 160–260 are similar to previous programs in that the catalog is memorized. See all the dull details in prior programs.

Now things get exciting in...

Lines 270–400, because we now use the previously collected material known as N$( ) and mix up the letters within each title. This uses the same technique used to shuffle cards, as described by David A. Lien about a hundred years ago when I had a TRS-80 Level I fully loaded with 4K. Thanks, Dr. Lien, I knew someday that card shuffling routine would help. It never did me any good at Caesar's Palace.

Line 410 is a familiar one, and in this was quite welcome I'm sure.

This program may be saved as Renamer.

Catalog Sector Count

I am sure by now several of you are howling with glee. For the majority of the readers—the majority of the minority who have lasted to this point—I feel that I must present one program of intrinsic value. I am sure it is intrinsic;

SECTOR COUNT

```
100  TEXT : HOME : CLEAR
110  DIM A(16),N$(30)
120  REM

*===================*
CATALOG SECTOR COUNT
*===================*

130  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NEVADA 89121

140  REM   IX/XIIII/MCMLXXXII

150  FOR I = 1 TO 16: READ A(I): NEXT I
160  PRINT  CHR$ (4)"CATALOG"
170  FOR X = 1 TO 16
180  FOR Y = 3 TO 8
190 N$(X) = N$(X) +  CHR$ ( PEEK (A(X) + Y) - 128)
200  NEXT Y
210 N =  VAL (N$(X))
220  IF  MID$ (N$(X),2,1) =  CHR$ (160) THEN 250
230 GT = GT + N
240  NEXT X
250  PRINT "   000"
260  VTAB  PEEK (37): HTAB (7 -  LEN ( STR$ (GT))); PRINT  STR$ (GT);" TOT
     AL SECTORS USED": PRINT "   ";493 - GT;" SECTORS STILL FREE ON THIS D
     ISK";
270  END
280  REM

===
ADD
===

290  VTAB 5: HTAB 1: CALL  - 958
300  PRINT GT
310  END
320  DATA  1536,1664,1792,1920,1064,1192,1320,1448,1576,1704,1832,1960,110
     4,1232,1360,1488,1616,1744,1872
```

I just looked it up in my Merriam-Webster Ex-Collegiate dictionary.

It is well known that only a limited amount of space is available on the Apple disk. DOS, according to the manual, stands for Disk Operating System, which requires the use of disk drives such as the Apple Disk II. I will not get into the relative merits of 35-track vs. 40-track, nor 5¼-inch vs. 8-inch disks. And using the other side of the disk is a whole new ball game.

> "It's likely you can deduct the cost of your computer and half your house rent by using this program."

Enough that each disk can hold only a certain amount of information. By counting the sectors (the numbers after the A (Applesoft) or I (Integer) or B (whatever that stands for) or any other strange letters on *your* catalog), you will know how much room is left.

Or, you can wait for the DISK FULL notice from your computer, wise guy. (In some states the preceding statement may have to end with the words "gal" or "person.")

This program solves that problem very nicely and is certainly educational. It's likely you will be able to deduct the cost of your computer and half of your house rent by using this program. Please consult with your financial adviser, tax consultant or Apple dealer for laws covering this situation.

Lines 100–150 clear screen and load data into the place where data gets loaded.

I really don't like using the variable I because it looks so much like a 1, so I use it for stuff like loading data and then don't worry about using it again for anything else. If I wrote things down when I programmed, I wouldn't use I as a variable at all.

Line 160 displays the catalog so we can go to work on it.

Lines 170–240 read the catalog, with the help of the numbers at line 320, which enable us to read the lines in the "correct" order. Surely, someone had something in mind when the sequence was decided in the manner in which it was. I really don't want to know about it.

Note that during the collection of information, line by line, only the characters at HTABS 3–8 are being collected. This is where the "sector numbers" are, on screen. On your disk they are in some mysterious place with hex

numbers all around them.

Line 210 gets the value of the array N$(X) and assigns it to N. Then, in line 230, GT accumulates the grand total of the numbers.

Lines 260–310 wrap up the program and the total number of sectors indicated in line 260. For some reason, not revealed to us here in the West, if you add up all the numbers and subtract them from 493 you will know (about) how many sectors are left.

This program should be saved under the file name Inventory.

Speed Lock

Here's a quickie for you folks who would like to try a program without peeks and pokes and smart aleck stuff.

This program will instantly lock or unlock all of your files. No big deal, but it is impressive if you have a neighbor looking over your shoulder. You know the one, he beats you in Pac-Man, Donkey Kong and all the other neat stuff that won't work on your Apple. Show him this and he'll never bother you again. Or even talk to you.

It is quite self-explanatory and only line 160 needs to be explained. The CHR$(8) is a backspace and has an important reason for being there. Something about dressing up the plain white box that keeps flashing all the time on my computer.

Save this program as Speed Lock.

SPEED LOCK

```
100  TEXT : HOME : CLEAR
110  REM

==========
SPEED LOCK
==========

120  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NV 89121


130  REM  IX/VII/MCMLXXXII

140  PRINT  CHR$ (4)"CATALOG"
150 T =  PEEK (37)
160  INVERSE : VTAB 1: HTAB 1: PRINT "(L)OCK  (U)NLOCK  (N)ORMAL   (Q)UIT
     ";: NORMAL : PRINT " ?"; CHR$ (8);: GET AN$
170 Z$ = " "
180  IF AN$ = "L" THEN Z$ = "*"
190  IF AN$ = "Q" THEN 250
200  IF AN$ = "N" THEN 260
210  FOR X = 5 TO T
220  VTAB X: HTAB 1: PRINT Z$
230  NEXT X
240  GOTO 160
250  VTAB 23: HTAB 1: END
260  PRINT : HOME
270  PRINT  CHR$ (4)"CATALOG"
280  GOTO 160
```

Disk Namer

I have two kinds of disks. Most of them are "protected" and so I have no idea in the world what is on them. I

never use most of them, since I'm afraid that if I *do* get started, and find them useful or fun, and something goes wrong, I'll never get a replacement set. After all, it took me five days to get my Apple fixed and it only had a stuck key on the keyboard. What would happen if I had to send the disk 2000 miles away to have it replaced?

The other kind I have all say VOLUME 254, or sometimes, when I feel like a real roué, VOLUME 253.

Disk Namer will bring a new light to your life. You will be able to see what a disk *can* look like, if you can get enough money together to buy one of those real disk naming programs.

Any good over-the-counter, non-prescription, non-addictive, safe relaxant is good to take before running this program to insure your blood pressure and pulse do not go completely nutzo.

Lines 100–150 clear the screen and establish a value for B$.

Line 160 determines the name to be used.

Line 170 determines mode of print.

Lines 180–280 print the name, verify text, and, if OK, the program ends.

Lines 290–320 comprise a subroutine that determines the print style for the name.

This program should be saved under the name Personalizer.

DISK NAMER

```
100  TEXT : HOME : CLEAR
110  REM

==========
DISK NAMER
==========

120  REM

PAUL RAYMER

130  REM

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NEVADA 89121


140  REM  IX/XIV/MCMLXXXII

150 B$ = "*"
160  PRINT "WHAT NAME WOULD YOU LIKE ON THE DISK": PRINT : INPUT "";A$
170  INPUT "(N)ORMAL (I)NVERSE (F)LASH ";S$: CALL  - 936
180  FOR X = 1 TO 40: VTAB 6: HTAB X: PRINT B$: VTAB 10: HTAB X: PRINT B$:
     NEXT X
190  VTAB 8: HTAB 3
200  GOSUB 290
210  VTAB 15: PRINT "IS THAT CORRECT? (Y/N) ";: GET AN$
220  IF AN$ = "Y" THEN 240
230  GOTO 100
240  PRINT : HOME
250  PRINT  CHR$ (4)"CATALOG"
260  VTAB 3: HTAB 1: CALL  - 868
270  GOSUB 290
280  VTAB 23: HTAB 1: END
290  IF S$ = "N" THEN  PRINT A$
300  IF S$ = "I" THEN  INVERSE : PRINT A$
310  IF S$ = "F" THEN  FLASH : PRINT A$
320  NORMAL : RETURN
```

REPLACER

```
100  TEXT : HOME : CLEAR
110  REM

========
REPLACER
========

120  REM

PAUL RAYMER

PAUL'S ELECTRIC COMPUTER
3464 TOWNHOUSE DRIVE
LAS VEGAS NEVADA 89121

130  REM  IX/XVIII/MCMLXXXII

140  PRINT
150  PRINT  CHR$ (4)"CATALOG"
160  VTAB 2: HTAB 1: PRINT "REPLACE WHAT LETTER (A-Z)? ";: GET AN$
170  VTAB 2: HTAB 1: CALL  - 868: PRINT "WITH WHAT CHARACTER? ";: GET C$
180  VTAB 2: HTAB 1: CALL  - 868
190  FOR X = 1024 TO 2038
200 L =  PEEK (X)
210  IF  ASC (AN$) + 128 = L THEN  POKE X, ASC (C$) + 128: PRINT  CHR$ (7)
     ;
220  NEXT X
230  VTAB 23: HTAB 1: PRINT " MORE? (Y/N) ";: GET AN$
240  IF AN$ <  > "N" THEN 160
250  VTAB 23: HTAB 1: CALL  - 868: END
```

Replacer

This is a very short program which is really more complex than it looks. It enables you to replace any letter on the screen with any other letter.

The value of this is immediately apparent. You may correct spelling errors on your disks. You may try out new program names. You can use dirty words and if you don't accidentally turn your printer on, no one will know.

You can also see how simple it is to find out what is on the screen (by *peeking*) and how easy it is to put the same thing or any other thing back on the screen by *poking* (POKE commands).

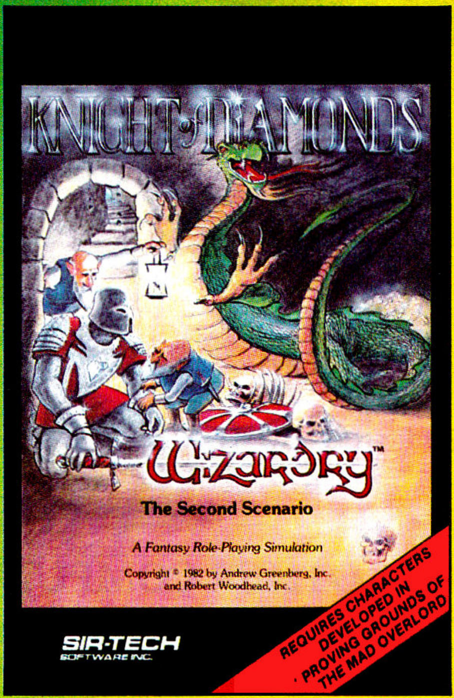
Neat formatting is nice.

Surely there is more to computing than Accounts Receivable, General Ledger, Inventory and Sales Projection.

Try the Replacer and see what I mean!

Hope you enjoy your Apple as much as I! ■

Wizardry



KNIGHT OF DIAMONDS . . . THE SECOND SCENARIO

The characters with whom you fought through to the 10th level in "Proving Grounds" must now call upon every ounce of resourcefulness to save the City of Llylgamyn. In fact, they must have reached the 13th level of power merely to survive!

The same high resolution, 3-dimensional visual effects that thrilled you in the first scenario are here—and more. "Knight of Diamonds," so eagerly anticipated by Wizardry players everywhere, surpasses highest expectations for story, action and complexity.

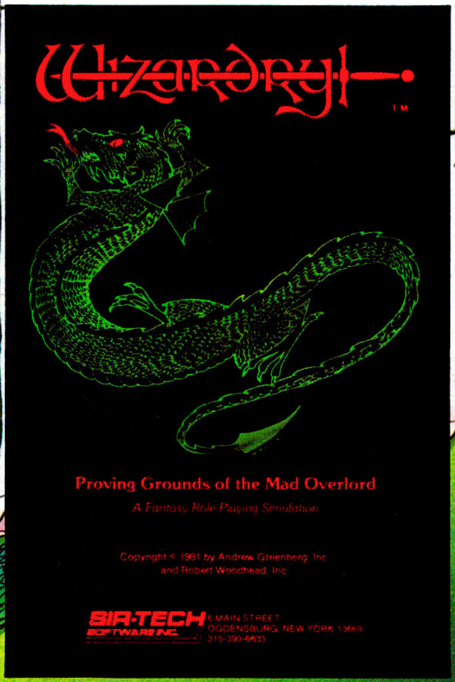
If successful, you will return the enchanted staff of Gnilda to Llylgamyn and *become* the Knight of Diamonds, but formidable adversaries block you at every turn. To begin your quest, simply **place yourself under the spell of Wizardry®.**

THE LEGENDARY FANTASY ROLE-PLAYING SIMULATION

. . . In "Proving Grounds of the Mad Overlord" you begin a challenging and totally absorbing journey through a 10-level, 3-dimensional maze. Create, then command, a hearty band of up to six adventurers who all must cooperate to explore a deep and mysterious maze in search of loot and glory.

The mages cast spells; thieves plot to steal treasure; and warriers battle the monstrous fiends of the underworld. In Wizardry®, no one remains unchanged; each member of the party grows in age, experience and, you hope, wisdom.

The puzzles, passageways and personalities in "Proving Grounds" have already fascinated and astounded the computer world and created a dedicated following of players. Find out why—**place yourself under the spell of Wizardry®.**



ALL SOFTWARE AVAILABLE FOR THE APPLE AND IBM PC AT YOUR FAVORITE RETAILER

SOFTWARE THAT COMPLIMENTS YOUR INTELLIGENCE

SIR-TECH SOFTWARE INC. 6 MAIN STREET
OGDENSBURG, N.Y. 13669
(315) 393-6633

Apple is a registered trademark of Apple Computer, Inc. IBM is a registered trademark of International Business Machines Wizardry is a registered trademark of Sir-tech Software, Inc.

Circle 218 on Reader Service card.

IT WORKS!

Have you pulled a fast one on your computer today? Sometimes a seemingly simple problem requires a little extra ingenuity—as this author discovered.

by Bob Blaske

Problem: How in the world could I get that machine to increment a counter for displaying a controlled amount of data... groups of 12, to be specific? How could this be done, pausing to await a user command and then continuing with the next display grouping, without an overabundance of If... Then... Gosub statements?

The sticker: You've got no idea how I sweated and grumbled and mumbled to myself as I sought the correct algorithm to solve the problem. An algebra teacher helped me trick my Apple into accepting this little program gem:

```
10 HOME
20 X=C+1
30 C=X
40 PRINT X,C
50 GOTO 20
```

Eureka, it works! Now here's the program, which is subroutine adaptable for those of us who are crippled without a printer to dump all that data to paper. Would you like groups of 12 in your screen display for copying a multi-year loan amortization, or other massive data output? You've got it:

```
10 Y=1
20 FOR X=0 TO 144
30 IF Y=X/12 THEN GOSUB 100
40 PRINT X+1
50 NEXT X
100 PRINT: INPUT "PRESS 'RETURN' ";A$
110 Z=0
120 Z=Y
130 Y=Z+1
140 ONERR GOTO 160
150 HOME: RETURN
160 END
```

Here's the point: You may be thinking, "Big deal! Why that's a simple programming skill. Anyone should be able to do that." Don't you see, that's my point—that's what *inCider* magazine is all about. I feel self-conscious about typing up this program and submitting it for publication for all the world to see. But maybe there's a quicker way to do the work and, if so, I want to know. If you know, share it with *inCider*. That's how we all get to learn. ■

Bob Blaske (8867 S.E. Colony St., Hobe Sound, FL 33455) is a professional educator, an amateur computerist, and is currently writing a historical novel in his spare time.

Fancy Listing Formatter

Tired of those unthinking folk who ruin your program listings with their three-hole punches? Fret no longer!

by Michael L. Waugh

```
0  POKE 33,33:PRINT CHR$(4);"OPEN LISTING":
   PRINT CHR$(4);"WRITE LISTING":LIST 1,32767:
   PRINT CHR$(4);"CLOSE LISTING":POKE 33,40:END
```

Listing 1. One-Line Capture program.

Listing 2. Indent Listing.

```
90  REM  *** PROGRAMMED BY MICHAEL WAUGH
100  REM  THIS PROGRAM IS SAVED AS --> INDENT LISTING (EPSON)
101 D$ =  CHR$ (4)
102  PRINT D$;"NOMON C,I,O"
110  ONERR  GOTO 999
230  HOME
240  PRINT "THIS PROGRAM LISTS ";: FLASH : PRINT "TEXT";: NORMAL : PRINT " FILES"
241  PRINT "IN INDENTED FORMAT"
242  PRINT "ON THE MX-80 PRINTER
243  PRINT : PRINT
250  PRINT "ENTER THE NAME OF THE ": INVERSE : PRINT "TEXT FILE";: NORMAL : PRINT " TO
     BE LISTED": INPUT "--> ";F$
251  IF  LEN (F$) > 40 THEN  HOME : PRINT "NAME TOO LONG": PRINT : PRINT : GOTO 250
252  PRINT : PRINT "ENTER A NAME FOR THE LISTING": PRINT "(30 CHARS OR LESS)": INPUT "--> ";N$
253  IF  LEN (N$) > 30 THEN  HOME : PRINT "NAME TOO LONG": PRINT : PRINT : GOTO 252
260  PRINT : INPUT "ENTER THE NUMBER OF LINES PER PAGE (MAX=50)--> ";N: PRINT
261  IF N > 50 THEN  HOME : PRINT "TOO MANY LINES": PRINT : PRINT : GOTO 260
280 P = 1:T = 15: REM    PAGE & TAB
290  VTAB 22: PRINT  TAB( 12);: FLASH : PRINT "TURN PRINTER ON": NORMAL : PRINT  TAB( 13);"PRESS
     ANY KEY"
291  GET Z$: IF Z$ = "" THEN 291
294  PRINT D$; CHR$ (1)
295  PRINT D$;"PR#1"
296  PRINT  CHR$ (18) CHR$ (14); TAB( 4);N$; CHR$ (14); TAB( 39);P
297  PRINT : PRINT  CHR$ (15): PRINT D$;"PR#0": IF P > 1 THEN 341
300  PRINT D$;"OPEN ";F$
341  PRINT D$;"READ ";F$
350 FL$ = ""
351  GET C$
352  IF C$ =  CHR$ (13) THEN 355
353 FL$ = FL$ + C$
354  GOTO 351
355  PRINT D$; CHR$ (1): PRINT D$;"PR#1"
```

Listing continued.

Anyone who spends hours programming what begins as "a simple little routine" knows that when they finish it's good to make several back-up copies "just in case." One back-up copy you should always make is a simple printed listing of the program. To do this for many programs, you just turn on the printer and type LIST. However, for special programs that you might want to catalog in a three ring binder, share with a friend or associate, or submit to a journal for publication, a little something extra is in order.

The Indent Listing program formats Basic program listings you plan to use for special purposes. It indents lines as many spaces as desired to accommodate 3-hole punching; it prevents split words at the ends of listed lines; and it titles and numbers the pages of your listing. In addition, it indicates when the end of the listing has been reached.

Indent Listing is written for use on Epson printers with the GrafTrax enhancement. It is simple to use, although neither as simple nor as fast as a normal LIST.

First, the "special" program must be "captured" as a text file. The DOS manual (p. 76) offers a short program

Address correspondence to Michael L. Waugh, 401 College Circle, Athens, GA 30605.

Listing continued.

```
370  REM  ***SCREEN FORMATTING ROUTINE***
374 B = 132 - (2 * T + 10)
376 BZ = 132 - (2 * T + 10)
378 BB = 132 - (2 * T + 5)
384 L =  LEN (FL$)
386  IF L < BB THEN  PRINT  TAB( T);FL$:KK = KK + 1: GOTO 500
388  FOR I = B TO BB
390 DU$ =  LEFT$ (FL$,I)
392  IF  RIGHT$ (DU$,1) =  CHR$ (32) THEN C = I + 1:I = BB: NEXT I: GOTO 398
394  NEXT I: IF B < 0 THEN  PRINT FL$:KK = KK + 1: GOTO 500
396 B = B - 5: GOTO 388
398 B = BZ: PRINT  TAB( T);DU$:KK = KK + 1
400  IF L - C > BB THEN 406
402  PRINT  TAB( T + 5); MID$ (FL$,C):KK = KK + 1
404  GOTO 500
406  FOR I = B TO BB
408 DU$ =  MID$ (FL$,C,I)
410  IF  RIGHT$ (DU$,1) =  CHR$ (32) THEN C = C + I:I = BB: NEXT I: GOTO 416
412  NEXT I
414 B = B - 5: GOTO 406
416 B = BZ: PRINT  TAB( T + 5);DU$:KK = KK + 1
418  GOTO 400
460  GOTO 500
470  REM  *** SKIP PERFORATION ***
480  FOR J = 1 TO 63 - N: PRINT : NEXT J
490  RETURN
500  IF KK >  = N THEN  GOSUB 470:KK = 0:P = P + 1: GOTO 296
501  PRINT D$;"PR#0": GOTO 341
999  PRINT D$;"CLOSE ";F$:Y =  PEEK (222): IF Y = 5 THEN  HOME : VTAB 12: PRINT "NORMAL EOF
     ENCOUNTERED": PRINT D$;"PR#1": PRINT  CHR$ (18): PRINT  TAB( 7);"END OF PROGRAM LISTING": PRINT
     D$;"PR#0": END
1000  PRINT D$;"PR#1": PRINT "ERROR # ";Y;" HAS OCCURRED.": PRINT D$;"PR#0": HOME : VTAB 12:
     FLASH : PRINT "ERROR # ";Y;" HAS OCCURRED.": NORMAL : END

END OF PROGRAM LISTING
```

to do this. Add the capture lines to the special program before running it. If the capture sequence occupies the first few lines of the special program, then only those lines are executed by the RUN.

If you want to avoid adding several lines to a program each time you list it, type in and save Listing 1, then merge it (using the Renumber program on the 3.3 system master) with your special program when you are ready to capture it as a text file.

With the special program captured, execute the formatted listing by typing RUN INDENT LISTING (EPSON) and, when prompted, enter the name of the text file version of your program. That's all there is to it, except for waiting—the indented listing takes two or three times longer to print out. This is because the Apple must Get strings from the disk one character at a time, then search for an appropriate place to break the line.

The Indent Listing program should lend itself readily to modification for other printers. To facilitate this process, several key lines and variables are listed below.

- Line 260. Variable N is the number of lines per page. Because of the heading information in this version, it has a maximum of 63. I set 50 as maximum (lines 260 and 261) for a nice bottom margin.
- Line 280. Variable T is the tab setting. It specifies the number of spaces the listing is to be indented.
- Lines 296 and 297. CHR$ codes 14, 15, 18 are Epson specific.
- Lines 374–378. Variables B and BZ should have the same value. They define the leftmost position of the initial target window, which is that portion of a printed line searched for as a logical breaking point (a space). Variable BB is the maximum printed line length. These values are currently set to center the listing on the page with right and left margins of T spaces. If this is not desirable, they could be set, for example, as BB = 40:B = 35:BZ = 35. This would set the line length to 40 characters maximum, with an initial target window of five characters.
- Lines 370–460. This is the formatting routine that prevents a word from being split at the end of a line. Variable FL$ must be defined, then the routine is entered at line 370. Exits from the routine are GOTO500 statements. Variable KK counts the number of lines that have been printed by the printer. This formatting routine can also be modified for printing strings to the Apple screen.

Listing 2 is a listing of the Indent Listing program running on itself. It provides an excellent example of the program's capabilities. ■

Interaction – A Child's World

by Tobi Hoffman

Interaction is a new department devoted to programs for and/or by youngsters. This one will delight preschoolers.

Keyboard Pix

Listing. Keyboard Pix program.

```
10  DIM L1$(61),L$(61,7),B$(31)
20  GOSUB 1000
30  HOME
40  GR
50  VTAB 21: GET A$: PRINT A$
60  GR : REM TO ERASE LAST LETTER
70 COL =  INT ( RND (1) * 16): IF COL = 0 THEN 70
80  COLOR= COL
90 X =  ASC (A$) - 33
100  IF X < 0 THEN 50
110  IF X > 62 THEN 50
120 R =  INT ( RND (1) * 35):S =  INT ( RND (1) * 33)
130  FOR Z = 1 TO 7
140  FOR Y = 1 TO 5
150  IF  MID$ (L$(X,Z),Y,1) = "1" THEN  PLOT Y + R,Z + S
160  NEXT : NEXT
170  GOTO 50
180  END
1000  REM  BINARY DATA
1005  FOR I = 1 TO 31: READ B$(I): NEXT
1010  DATA 00001,00010,00011,00100
1011  DATA 00101,00110,00111,01000
1012  DATA 01001,01010,01011,01100
1013  DATA 01101,01110,01111,10000
1014  DATA 10001,10010,10011,10100
1015  DATA 10101,10110,10111,11000
1016  DATA 11001,11010,11011,11100
1017  DATA 11101,11110,11111
1020  FOR X = 0 TO 61
1022  READ L1$(X)
1024  FOR Z = 1 TO 7
1026  READ B:L$(X,Z) = B$(B)
1028  NEXT : NEXT
1030  RETURN
1031  REM  PICTURE DATA
1033  DATA "!",4,4,4,4,4,0,4
1034  DATA "''",10,10,10,0,0,0,0
1035  DATA "#",0,10,31,10,31,10,0
1036  DATA "$",4,15,20,14,5,30,4
1037  DATA "%",25,26,4,4,8,11,19
1038  DATA "&",12,20,8,12,21,18,13
1039  DATA "'",12,12,4,0,0,0,0
1040  DATA "(",2,6,4,8,4,6,2
1041  DATA ")",8,12,4,2,4,12,8
1042  DATA "*",4,21,14,31,14,21,4
1043  DATA "+",0,4,4,31,4,4,0
1044  DATA ",",0,0,0,0,6,6,2
1045  DATA "-",0,0,0,31,0,0,0
1046  DATA ".",0,0,0,0,6,6,0
1047  DATA /,0,1,2,4,8,16,0
1048  DATA 0,14,10,17,17,17,10,14
1049  DATA 16,4,12,4,4,4,4,14
1050  DATA 2,14,17,19,2,4,8,31
1051  DATA 3,14,19,1,14,1,19,14
1052  DATA 4,6,10,18,31,2,2,2
1053  DATA 5,31,16,16,30,3,17,14
1054  DATA 6,14,25,16,22,25,17,14
1055  DATA 7,31,17,2,4,4,8,8
1056  DATA 8,14,17,10,4,10,17,14
1057  DATA 9,14,17,19,13,1,19,14
```

Listing continued.

Adam, age three, has been busy for the last 15 minutes with this little program (see program listing) that responds to any keypress with a lo-res character, randomly located on the screen and in random colors. He has learned the vital lesson of the Apple reset button, and calls the escape key an "off" button because it erases the screen.

The punctuation keys puzzle him, as they do not fit into the categories of numbers or letters, which he knows.

Ah, now he has discovered the repeat key! Together with the O, it makes the computer draw big O's on the screen again and again. He doesn't seem to realize that any key could be repeated.

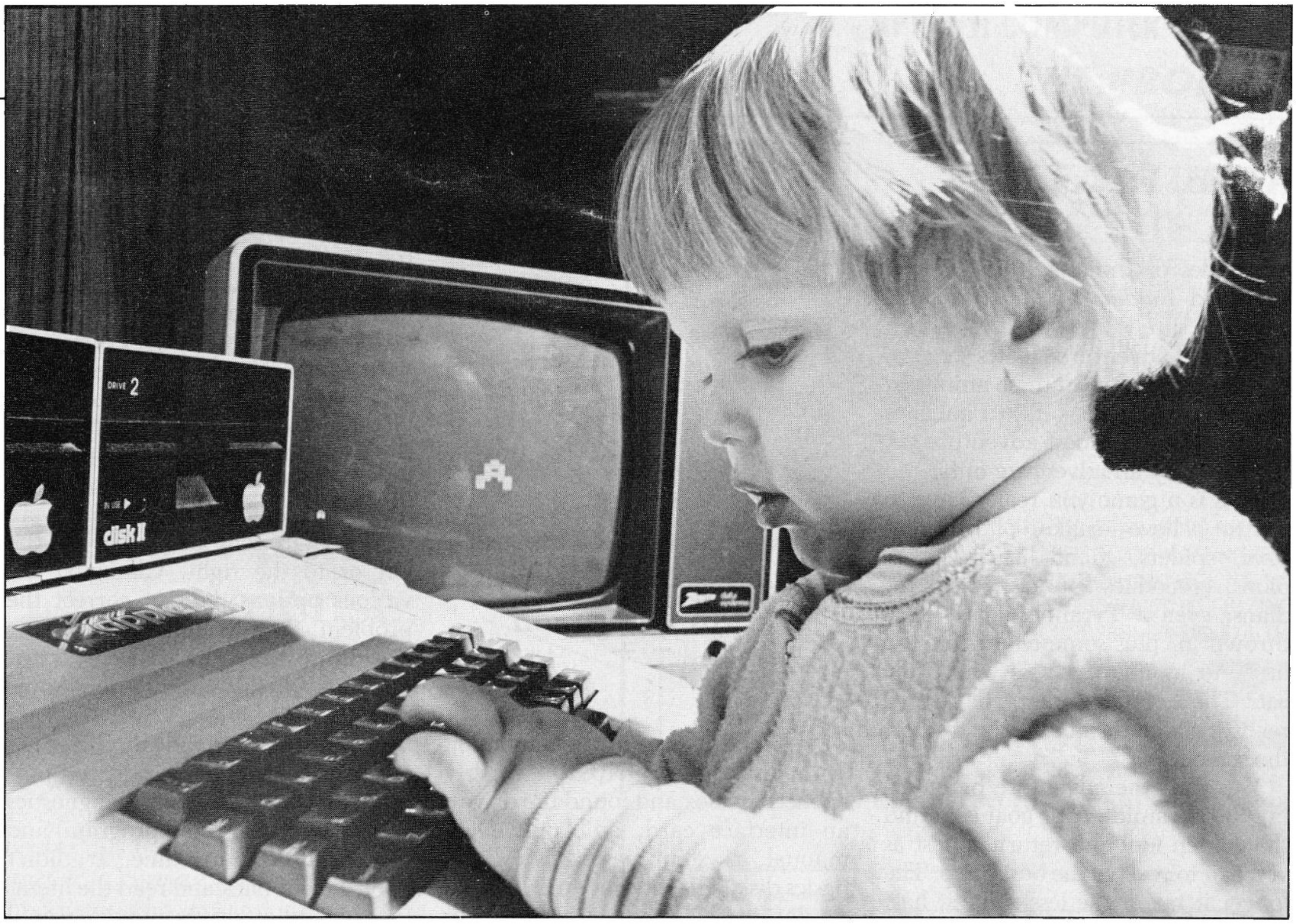
The shift key affects the number keys and a few others, and learning how takes a bit longer. (I wish it could be used for upper/lowercase, too, but I haven't put that wire in.)

After half an hour, now, the game has begun to wear thin—oops, no it hasn't! He just found out that the space bar is also an "off" button!

I tried this program out on another three-year-old the other day, and regretted doing so as he began to pound the keyboard in typical toddler style. Adam, however, has been very gentle, using one finger or, frequently, his thumb, to press the keys. With small children, the rule must be quite clear how one treats the computer: One transgression means no more play.

Another reset! I could program

Address correspondence to Tobi Hoffman, 58 Hilldale Road, Ashland, MA 01721.



Child's play!

that out, but it does teach a good lesson—and not just to future programmers. In every game there are rules to abide by—and this one is self-enforcing.

The pictures are generated with a binary-type code; the array L$(x,y) is filled with seven lines which are the binary equivalents of 1 to 31. A 1 is plotted, a 0 is not. The data read in is in decimal form, and is converted to binary. I originally had the binary codes assigned directly, but found that reducing them to decimal figures simplified the program entry.

Those of you with ambition and free time could expand this program to include pictures with the letters, or to be a mini- (micro?) word processor. I once had in mind to allow several letters to be printed in rows, and include a list of words that the program would recognize. If such a word were entered, the picture would be drawn. The child's own name could be put into this list. I would like to hear from anyone who tries his/her own adaptations. ■

Listing continued.

```
1058  DATA ":",0,12,12,0,12,12,0
1059  DATA ";",0,12,12,0,12,12,4
1060  DATA "<",0,2,4,8,4,2,0
1061  DATA "=",0,0,31,0,31,0,0
1062  DATA ">",0,8,4,2,4,8,0
1063  DATA "?",14,17,2,4,4,0,4
1064  DATA "@",14,17,21,23,18,16,15
1065  DATA A,4,14,10,14,27,17,17
1066  DATA B,28,18,22,28,22,18,28
1067  DATA C,14,25,16,16,16,25,14
1068  DATA D,28,18,17,17,17,18,28
1069  DATA E,31,16,16,30,16,16,31
1070  DATA F,31,16,16,30,16,16,16
1071  DATA G,14,25,16,16,23,18,14
1072  DATA H,17,17,17,31,17,17,17
1073  DATA I,14,4,4,4,4,4,14
1074  DATA J,2,2,2,2,2,18,12
1075  DATA K,17,18,20,24,20,18,17
1076  DATA L,16,16,16,16,16,16,30
1077  DATA M,27,27,21,21,17,17,17
1078  DATA N,17,25,25,21,19,19,17
1079  DATA O,14,27,17,17,17,27,14
1080  DATA P,30,19,17,19,28,16,16
1081  DATA Q,14,27,17,17,21,18,13
1082  DATA R,30,19,17,19,28,22,17
1083  DATA S,14,17,24,4,3,17,14
1084  DATA T,31,4,4,4,4,4,4
1085  DATA U,17,17,17,17,17,27,14
1086  DATA V,17,17,27,10,10,14,4
1087  DATA W,17,17,17,21,21,27,10
1088  DATA X,17,27,10,4,10,27,17
1089  DATA Y,17,10,10,4,4,4,4
1090  DATA Z,31,1,2,4,8,16,31
1091  DATA NULL,0,0,0,0,0,0,0
1092  DATA NULL,0,0,0,0,0,0,0
1093  DATA ],7,1,1,1,1,1,7
1094  DATA ^,4,10,17,0,0,0,0
2000  REM  LAST TWO DIGITS OF
      LINES 1033 TO 1094 ARE ASCII
      CODE OF CHARACTER
```

Aztec

Shades of Indiana Jones! If you liked *Raiders of the Lost Ark*, you will surely like Aztec. I've seen some good adventure Hi-Res screens, but none compare to these animated graphics. My hat's off to Paul Stephenson for the finest adventure yet (and I'm not an adventure buff).

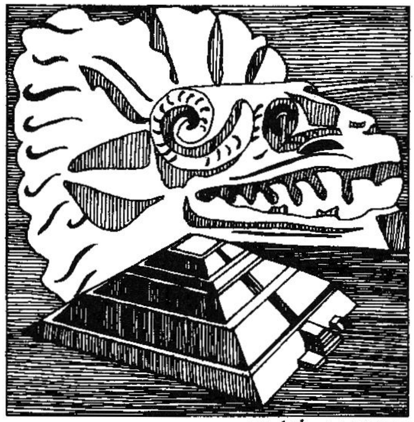
This is a game you really have to see to believe—snakes of different sizes, spiders, giant lizards, scorpions, crocodiles, pumas, Aztec Indians; even a Tyrannosaurus rex is thrown in, plus some characters too nasty to mention. Guess what they eat? The animated characters move realistically and smoothly on and off the screen.

You start the adventure by entering the pyramid. Your goal is to find the golden idol and return as fast as you can to receive the best price. The longer it takes, the less value it has. You can walk, run, jump, crawl, lunge and shoot, as well as plant explosives. Be careful you don't blow your bridges behind you as well as your own behind. There are many levels to explore, as well as many baskets and piles of debris. Weapons are hidden, as well as booby traps. If you're lucky enough to avoid or to kill the monsters around you, a booby-trapped room can start to close in on you. How do you escape from that? I'm not going to tell.

The commands can be seen by hitting the escape key at any time. All the commands are in the keyboard mode, so do your finger exercises before you play. It takes a while before the commands become second-nature. The fight key may seem the smallest key on your keyboard when you need it the most—also the hardest to find.

Aztec is a wonderful arcade game/adventure by Paul Stephenson and Datamost (9748 Cozycroft Ave., Chatsworth, CA 91311). It's an excellent value at $39.95. Happy adventuring! ■

George Engel
Seymour, CT



The Forth Dimension Disk Drive

At long last our Forth Dimension disk drives arrived! I eagerly opened the box and found the drives, an interface card, an Apple DOS manual, a system master disk, and a basics disk. The interface card looked similar to the Apple interface, but there are differences that became apparent on closer inspection. One difference is that the Forth card does not include all of the gold pin connectors. Maybe they are not used?

I connected the plain grey cables to the interface and installed the drives in the Apple. Booting a system master was quiet, with only a few switches clicking and the sound of an industrial quality motor turning the disk. The programs ran fine, and the drive sounds were quiet and reliable.

I then removed the cover to look inside, and to find the speed control. I found the speed control on the bottom of the unit—easily accessible from the outside. Removing the bottom cover from the drive revealed a belt drive mechanism that looks similar to the one on the Apple drive. There are strobe marks on the large drive wheel so you can set the speed. There is a screw about 3/8-inch in diameter, turned by a stepper motor, that positions the head over the disk. This looks professional compared to the plastic cam used by Apple.

The Forth drive is made by Siemens, the Apple drive by Shugart. The two drives look quite different inside. The Forth drive has arms that appear to position the disk before the door is closed and a mechanism that will not let you close the door unless the floppy is correctly positioned!

At this point I reassembled the drive. The strobe showed the speed to be adjusted correctly, but I thought it wise to check the drive speed with some software. I booted Locksmith and selected the drive speed option; the result was somewhat confusing. Many of the lines were as they appear on the Apple drive, but there were several that went all the way to the left or to the right. Changing the various options did not correct the problem. You could probably use Locksmith to set drive speeds, but it doesn't look the same as the Apple drive display.

Because of the erratic display, I decided to use another disk speed program. I inserted Master Diagnostics Plus (a copy-protected program) and turned on the machine. It didn't boot. I went back and read the literature on add-on drives and discovered the problem: Most of the add-ons are billed as being 100 percent compatible with CP/M, Pascal and DOS 3.3; however, if you have a copy-protected foreign DOS, you take your chances.

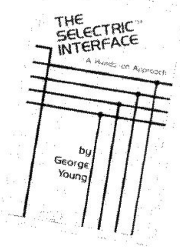
Wizardry was another program that did not run on the Forth. Although the program appeared to run, I could not assemble my party. It is yet another program that uses extremes in its copy-protection scheme.

Programs that are copy protected and run adequately include: Locksmith, Pilot, and Apple Diagnostics. DOS 3.3 files also run fine on the Forth. Thus far, the only problems I have encountered have been with Master Diagnostics Plus and Wizardry. But it is disturbing not to know what will definitely run.

The Forth drive seems to have much-reduced startup-switchoff times as compared to the Apple. This is a timesaver for text files and Pilot.

The Forth Dimension drive may have 40-track capability, but there is no software or documentation supplied with the drive (other than the DOS manual), so the capability cannot be easily used.

THE SELECTRIC™ INTERFACE



Daisy wheel quality without daisy wheel expense.

You need the quality print that a daisy wheel printer provides but the thought of buying one makes your wallet wilt. *The Selectric™ Interface*, a step-by-step guide to interfacing an IBM Selectric I/O Writer to your microcomputer, will give you that quality at a fraction of the price. George Young, co-author of *Microcomputing* magazine's popular "Kilobaud Klassroom" series, offers a low-cost alternative to buying a daisy wheel printer.

The Selectric™ Interface includes:

- step-by-step instructions
- tips on purchasing a used Selectric™
- information on various Selectric™ models, including the 2740, 2980, and Dura 1041
- driver software for Z80, 8080, and 6502 chips
- tips on interfacing techniques

With *The Selectric Interface* and some background in electronics, you can have a high-quality, low-cost, letter-quality printer. Petals not included.

Credit card orders call TOLL-FREE 1-800-258-5473. Or mail your order with payment plus $1.50 shipping and handling to: Wayne Green Inc. Attn: Retail Book Sales, Peterborough, NH 03458.

Dealer inquiries invited.

ISBN 0-88006-051-4 128 pages **$12.97**

☐ Yes, I want **Selectric Interface** (BK7388). Enclosed is $12.97 per copy plus $1.50 for shipping and handling.

☐ MASTER ☐ VISA ☐ AMEX

Card # ______________ Expires ______________

Signature ______________

Name ______________

Address ______________

City ______________

State and Zip ______________

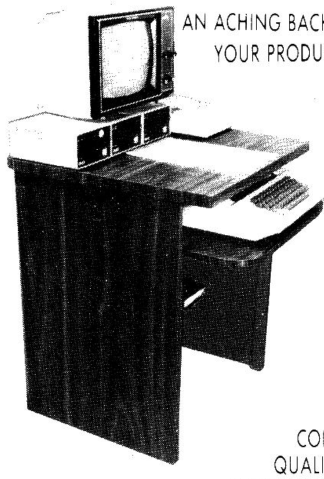
All orders shipped UPS if complete street address is given.

334B5S

Circle 57 on Reader Service card.

WOW! A FUNCTIONAL, COMFORTABLE APPLE II COMPUTER STATION

THE PROBLEM:



AN ACHING BACK RESTING ON A TIRED "*" REDUCES YOUR PRODUCTIVITY, CREATIVITY AND ENJOYMENT.

THE SOLUTION: BASIC COMFORT II

DESIGNED BY A DEDICATED COMPUTER HOBBYIST, IT POSITIONS THE KEYBOARD AT A HEIGHT THAT REDUCES FATIGUE AND ACHES OF THE LOWER BACK AND SHOULDERS. WITH DISK DRIVES AND CRT IN PLACE THERE ARE ABOUT TWO SQUARE FEET (15" X 25") OF WORK AREA DIRECTLY IN FRONT OF YOU FOR READ'N AND WRIT'N. CONSTRUCTED OF ALL WOOD FURNITURE QUALITY PARTICLE BOARD AND FINISHED IN WOODGRAIN (WALNUT OR OAK) HIGH PRESSURE MICA LAMINATE, BASIC COMFORT II IS RUGGED (70 LBS.) TO ORDER OR INQUIRE PHONE **1-800-874-3514**

IN FLORIDA CALL COLLECT (904) 252-7970

$169.00

master charge VISA

PICTURE HOUSE

BASIC COMFORT II COMPUTING PRODUCTS

166 BOYNTON BLVD., DAYTONA BEACH, FL 32018

Circle 53 on Reader Service card.

DISK DRIVE WOES?
PRINTER INTERACTION?
MEMORY LOSS?
ERRATIC OPERATION?

Don't Blame The Software!



Power Line Spikes, Surges & Hash could be the culprit! Floppies, printers, memory & processor often interact! Our patented ISOLATORS eliminate equipment interaction AND curb damaging Power Line Spikes, Surges and Hash. MONEY BACK GUARANTEE!

- **ISOLATOR (ISO-1) 3 filter isolated 3-prong sockets; integral Surge/Spike Suppression; 1875 W Maximumm load, 1 KW load any socket** $76.95
- **ISOLATOR (ISO-2) 2 filter isolated 3-rpong socket banks; (6 sockets total); integral Spike/Surge Suppression; 1875 W Max load, 1 KW either bank** $76.95
- **SUPER ISOLATOR (ISO-3) similar to ISO-1 except double isolation & Suppression** $115.95
- **SUPER ISOLATOR (ISO-11) similar to ISO-2 except double isolation & Suppression** $115.95
- **MAGNUM ISOLATOR (ISO-17) 4 Quad isolated sockets; For ULTRA-SENSITIVE Systems** $200.95
- **CIRCUIT BREAKER, any model (Add-CB)** .... Add $10.00
- **REMOTE SWITCH, any model (Add-RS)** .... Add $18.00

AT YOUR DEALERS

MasterCard, Visa, American Express ORDER TOLL FREE 1-800-225-4876 (except AK, HI, PR & Canada)

ESP Electronic Specialists, Inc.

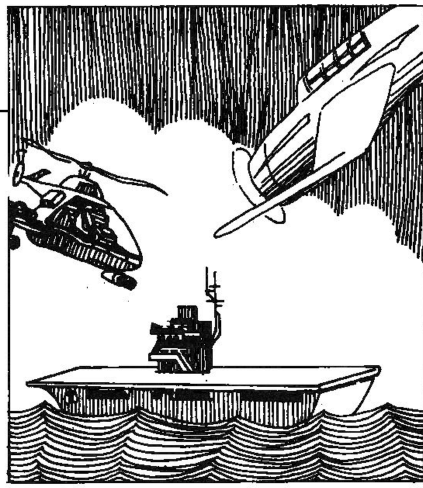
171 South Main Street, Box 389, Natick, Mass. 01760
(617) 655-1532

I am impressed with the Forth. It appears to be well engineered and is clearly not a toy. Its peculiar behavior is disturbing. However, I am getting to the point where I don't want a program in a protected format. I have seen these drives advertised for $299, and they seem to be an excellent value, especially as a second drive. Available from Siemens Corp., 240 E. Palais Rd., Anaheim, CA 92805. ■

David C. Lowe
Lexington, KY

Wavy Navy

Sirius Software, 10364 Rockingham Drive, Sacramento, CA 95827, has done it again. Wavy Navy, selling for $34.95, has excellent graphics and sound, and the music is good as well. Once the title page has been shown, you're offered an option screen that looks like Figure 1.



1) 1 PLAYER
2) 2 PLAYERS
3) 3 PLAYERS
4) 4 PLAYERS

5) BEGINNER
6) ADVANCED
7) EXPERT

8) APPLE SPEAKER
9) CASSETTE PORT

X) PADDLE 0
Y) PADDLE 1
Z) JOYPORT
K) KEYBOARD

D) DEMONSTRATION
S)START GAME

Figure 1. Option screen of Wavy Navy.

As you can see, you can choose how many players will play the game, the difficulty level (I think you'd better start out at the beginner's level), if you want sound or not, what kind of controls you want to use, and whether you want to watch a demonstration or start the game. (In case you don't know, the Joyport is a piece of equipment you can add to your Apple that lets you use either paddles or Atari joysticks.)

There are ranks in this game. You earn higher ranks by doing well. You start out as a galley slave, and you can go as high as President. In the beginning of the game, you have three ships. Each time you make it through a level, you get an extra ship.

Sound good so far? There's always something not-so-good waiting around the corner, though. You can get destroyed when any of the following happen to you:

- A kamikaze plane hits you
- A bomb hits you
- You get shot by a helicopter
- You run into a mine
- A rocket hits you

Now it doesn't sound so good. The kamikaze plane tries to smash right into you, as you ride up and down on the waves. This wave action also makes it hard for you to hit your target. Watch out for the helicopters—their machine guns are pretty accurate. The mines are under the ocean. When you're going up and down on a wave, try to stay on the wave and not in its trough. If the mine is in the lower part of the wave, and you're going down with the wave, the mine'll get you every time. The rockets are just something else you have to worry about.

The first level of play requires your full attention. Helicopters are machine-gunning you, and the kamikazes are all out to sink you. The second level adds mines to your troubles. The bad thing about the mines is that you can't shoot them. You just have to avoid them.

The third level is like the second, but here a bomber flies across the screen and drops about 20 bombs on you. I got through the third level with nerves of steel and tricky navigation.

Some convenience features with the game include the escape key, which pauses the game—another keypress restarts the game. Control-S toggles the sound between the cassette port and the Apple speaker. And last, but not least, the control-R ends the game and returns you to the main menu.

I thought Wavy Navy was excellent. Now if I can only get my Mom and Dad to let me play the game. Dad always did want to be President. ■

Kirk H. Lesser
Hancock, NH

SAM, Software Automatic Mouth

To be very honest, I've always felt something was lacking in my computer system. It has great color graphics, juggles numbers like a mainframe and does books better than a Wall Street accountant. Still, something was missing. It never talked.

Well, I'm happy to say that my machine is now quite capable of chatting away, thanks to SAM, the Software Automatic Mouth from Don't Ask Software, 1700 Pontius Ave., Suite 201, Los Angeles, CA 90025, 213-477-4514.

SAM is neither the first nor the most exotic speech synthesizer available but it is unique in several ways. Up until now, most speech synthesizers for home computers produced a voice totally devoid of feeling or expression. They sounded just like the people in low-budget 1950's science fiction films after the creature from Venus took over their minds.

SAM's programming options provide for changes in pitch, stress and inflection, as well as speed. For example, with the proper word stressed, the phrase, "I saw him go," could answer any of the following questions: "Who saw him go?" "Whom did you see go?" "What did you see him do?" and "Did you see or hear him go?"

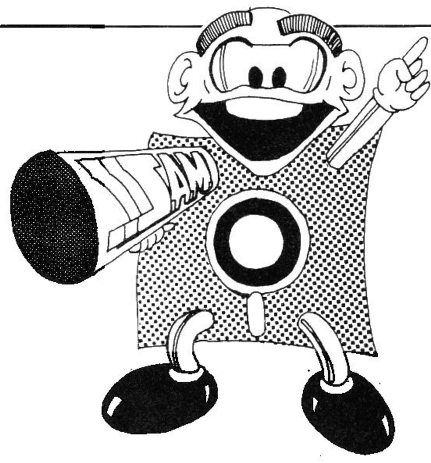
Also unique to SAM is the option

Circle 76 on Reader Service card.

for translating words into speech from normal English, rather than only from phonetic input. Of course, in this mode not every English word gets spoken accurately, but considering the complexity of the process, the result is amazing.

SAM's hardware is very simple—a D/A converter and a small audio amplifier, both mounted on a board approximately 3½ inches by 3½ inches. Installation consists of plugging the board into a slot on the Apple's main bus. SAM works in slot 4, but any slot can be used by poking the slot number into memory.

Once installed, SAM provides audio output via a two-conductor wire leading out of the computer for connection to a small loudspeaker you provide. The board contains a small potentiometer to set the volume level, and also gives you the option of bypassing the speaker in the Apple, amplifying signals normally produced there and playing them through the speaker used with SAM. I was pleased to discover that even with the relatively inefficient speaker I used, the on-board amplifier produced an impressive volume level.



All aspects of SAM are well explained in the superb user's manual provided with the board. Don't Ask has written one of the best manuals I have seen with any peripheral product. All of the support programs with SAM are thoroughly explained, and examples of interface to other programs are given. Part of the manual is dedicated to a dictionary for English-to-phonetic conversion. Although not every common word is listed, I can't think of anything that could not be expressed via various combinations of the words provided. All memory locations used with SAM are also listed in the manual, along with technical descriptions of the board itself.

On the main disk supplied with SAM there are several demo programs that give some interesting and amusing examples of SAM's versatility. A utility program, SAYIT, allows you to input information in either English or phonemes and to hear SAM say the word or phrase immedi-

ately—quite handy when developing a talking program.

Once the programs are loaded, it is easy to interface to SAM from your Basic or machine language program. From Basic, just assign the spoken phrase to a string variable, SA$, then call the SAM routine. The computer will then say whatever is written in the string and return. The possibilities for operator prompts and feedback seem endless.

From what I can tell, SAM can say just about anything well. With the proper inflection and choice of words, it could make a sailor blush.

Don't Ask Software has plans for the release of Singing SAM, a program to raise SAM's cultural level, and Poker SAM, to lower it again. I for one don't know if I could deal with having a computer take all my money at poker and then make fun of me.

The most surprising part of SAM for me was the price. The entire system, hardware and software (but no speaker), is only $124.95—very reasonable in the world of Apple add-ons.

Considering the price, the ease of use and the simplicity of installation, I deem SAM a good investment. And I'm hard pressed to think of anything else I've added to my computer system that has given me as much fun, or has made me think of as many possibilities for the future.

I've got to go now, my wife is calling me. . . . . . or is she? ■

Chuck Doherty
South Dartmouth, MA

Pest Patrol

Pest Patrol, from Sierra On-Line Inc. (36575 Mudge Ranch Road, Coarsegold, CA 93614; $29.95) is one of the most enjoyable arcade-type games I've seen. This game borrows some of the best points from Space Invaders, Galaxian Centipede and others to keep you hopping for hours.

It is difficult to design a game where there is enough challenge to keep you from boredom, but not so much that frustration prevents you from enjoying it. Sierra seems to have hit upon the right formula for this one. The graphics are bright, clean and fast moving; sound is appropriate to the action and can even be turned off when not wanted.

The game pits you, armed only with a can of insecticide, against a varied and unpredictable group of insect invaders. The challenge is increased by the bombs the insects drop as they attack. Different types of bugs require different strategies to destroy

them. Some can be destroyed with one whiff of your spray while others may take as many as five shots before you are rid of them. There are even insects that must be physically hit with the can before they will leave you alone.

Pest Patrol's insects act like the real thing. The fleas drop, jump, and drop again, leaving you to catch them unaware. The beetles fly in attack formation, like so many airplanes attacking a battleship. As play progresses and the difficulty increases, the attack patterns of the insects become harder to predict. To survive in this insect-filled jungle, you have to be on your guard every second.

Different levels of play can be selected, even mid-game (a nice feature), but there is also the unnerving policy of automatically lowering your play by one level whenever your can is destroyed. Forty different levels of difficulty are available (30 selectable, the last ten must be earned), but even the lowest level will keep you challenged the first time out. Fast and slow speeds keep the game at your chosen pace. The game has a practice mode and will go into a demonstration routine if left unattended.

This game will not bore you quickly; it is fast, challenging and quite unpredictable. Pest Patrol gets four stars in my computer game hall of fame. ■

Chuck Doherty
South Dartmouth, MA



Labyrinth

Bloody creatures everywhere! Just when you think you're safe, why, a door suddenly appears in the wall and you're face to face with one of THEM—a Trapper; or, even worse, a Scourge; or, even worse than worse, a Minotaur. The next thing you know, the game is over, with all four of your adventurers lying in the dust.

Labyrinth initially appears to be nothing more than a generic Snoggle game with a few different twists. There's a maze, and little dots that disappear as you move the blue-diamond playing piece over them. But, here the resemblance ends—quite quickly, as a matter of fact. For one thing, no joystick is employed here—only keyboard controls that require Olympic agility of your fingers.

The I, J, L and K keys control your diamond's motion up, left, right and down, respectively. In like fashion, E, S, F and D fire your weapon. The astute gamer will soon discover that coordinating movement and shooting requires two hands, or a single hand and a very adroit foot with above-average knee flexion.

In other words, these controls take considerable getting used to. The process is liable to cost several games, and, meanwhile, the inability to control the diamond is aggravating. But, is that all there is to the game, you ask?

No, indeed. The labyrinth is a maze of diamond mines, once owned by Prince Julian. Now abandoned, the mines harbor all manner of nightmarish creatures. And the walls themselves are constantly changing—merging, disappearing, sprouting lethal doors. They also hold four persons captive. Your task is to rescue them.

The prisoners are, of course, all in separate areas. Rescue all four, and you can dash for a flashing X that appears on screen. Enter the X and you go on to a higher level of play.

There are eight levels, each considerably more difficult than the last. The saving grace is that your player (defender) retinue increases by one at each new level attained.

The Trappers, the Scourges and the Minotaurs all shoot at you on sight. Even their touch will kill. Your single defense, alluded to above, is a trapper-zapper. For every creature you destroy, you are granted two additional shots. Every prisoner you free yields a single additional shot. To waste your shots is to court disaster. As the instruction card states "... life without zappers is like shark fishing without a boat..." Build up your ammunition supply, if possible. Then, if the walls ensnare you, you can blast your way out.

In some games, sit-and-wait-for-the-uglies works fairly well. Not so with Labyrinth. The fiendish folk figure out where you are if you move too slowly, and they come searching.

I find Labyrinth an excellent arcade game. Its superficial resemblance to Pac-Man rip[illegible] give pause. However, once past that, the ever-changing maze, the ferocity of the unsightly monsters, and the apparent hopelessness of the prisoners lend interest and excitement. It may well become a favorite.

Labyrinth comes from Broderbund Software Inc., 1938 Fourth St., San Rafael, CA 94901; (415) 456-6424. It requires an Apple II or Apple II Plus with 48K RAM and version 3.2 or 3.3 DOS. The price is $29.95. ■

Hartley G. Lesser
inCider staff

Circle 267 on Reader Service card.

NEW-ALTERNATIVES

IN SELECTING DISCOUNT SOFTWARE FOR YOUR APPLE

- user friendly service
- fairy tale prices
- special specials????

See the difference our catalog can make

VULCAN SOFTWARE

1805 SAULTER ROAD
BIRMINGHAM, AL 35209

(205) 871-5510

APPLE IS A REGISTERED TRADEMARK OF APPLE COMPUTER, INC.

Circle 250 on Reader Service card.

NOW SUPPORTS COLOR PRINTING!

TIRED OF ALL THE "EXCEPT FOR . . ."S?

WITH THE NEW PRINTOGRAPHER GRAPHICS PRINTING SYSTEM, WE'VE GOT YOU COVERED!

PRINTOGRAPHER™



PRICE: $49.95
(California Residents add 6% sales tax.)
Plus $2.50 Shipping

The **PRINTOGRAPHER** is designed to fill all of your graphic printing needs, without having to worry about running into the problem of it almost working, "except on your printer", or "except for the lack of that particular feature". Whether you have a daisy wheel or dot matrix printer, the standard version of **PRINTOGRAPHER** works on any printer and interface combination with graphics capabilities. In many cases, this includes printers you may not even have thought could print graphics.

Just a FEW of the possible printers include: EPSON, PAPER TIGER, APPLE DMP, PRISM, C. ITOH, GEMINI, BASE 2, ANADEX, NEC, DIABLO, QUME, MPI, SILENTYPE, OKIDATA, MALIBU; interface cards include: APPLE, SSM, CCS, MTN COMP. CPS, MPI, GRAPPLER, TYMAC, PROMETHEUS and more!

In addition to versatile print options (easy cropping, variable magnifications, normal/reverse inking, vertical/horizontal format, etc.) **PRINTOGRAPHER** offers such unique features as the ability to print pictures directly from disk (without loading a file), spooling via our **DOUBLETIME PRINTER** package, or sending pictures over a phone line using **ASCII EXPRESS**. You can even put graphics in your text documents with our text editor software, **THE CORRESPONDENT**. As if that wasn't enough, we've made it easy to put the **PRINTOGRAPHER** routines right in your own programs to do Hi-Res printing immediately during their operation, without having to save screen images to disk!

We also know you see a lot of advertising these days for a truly overwhelming volume of software, all claiming to be the best, so we make this simple guarantee:

IF YOU CAN FIND A BETTER PACKAGE THAN (OR ARE AT ALL UNSATISFIED WITH) THE PRINTOGRAPHER WITHIN 30 DAYS OF PURCHASE, SIMPLY RETURN THE PACKAGE FOR A COMPLETE REFUND, NO QUESTIONS ASKED!

For more information, see your local dealer, or write **SOUTHWESTERN DATA SYSTEMS** for a free catalog. If your dealer is out of stock we can ship **PRINTOGRAPHER** to him within 24 hours of a call to our offices.

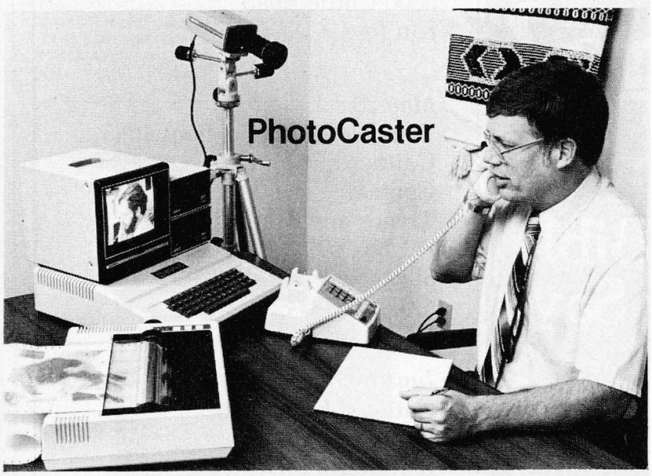
REMEMBER: WITH PRINTOGRAPHER, YOU'RE PICTURE PERFECT!

SYSTEM REQUIREMENTS: Apple II, II+, IIe

southwestern data systems™

P.O. Box 582-I • Santee, California 92071
Telephone: 619/562-3221

Circle 201 on Reader Service card.



PhotoCaster...a *new* feature packed system to take, process, store, print, send and receive color and black-and-white photos with your Apple II computer.

PC-100 (disk software, I/O board, manual, demo tape) **$499.95**

PC-101 (above plus Panasonic TV camera, RGB filter accessory) **$749.95**

Write or call for details

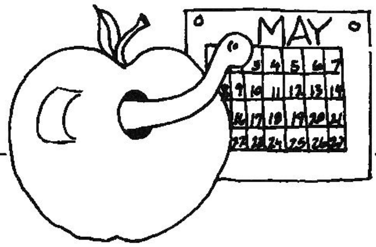
Visa and Mastercard orders accepted. Shipping charges and applicable California sales tax will be added.

Apple II TM Apple Computer, Inc.
PhotoCaster TM COMMSOFT, Inc.

COMMSOFT
2452 Embarcadero Way
Palo Alto, CA 94303

(415) 493-2184

Calendar



With this issue inCider *begins a calendar of coming events important to Apple users. If you are sponsoring, or know of, an event that should be included, please let us know. Write* **inCider** ***Calendar****, Pine Street, Peterborough, NH 03458.*

—the editors.

April 4–8
Tutorial Week East '83
Orlando, FL
Contact:
Tutorial Week East
PO Box 639
Silver Spring, MD 20901
(301) 589-8142

April 15–17
Anaheim Applefest
(exhibits)
Anaheim, CA
Contact:
National Computer Shows
822 Boylston St.
Chestnut Hill, MA 02167
(617) 739-2000

April 17–22
IEEE Infocom '83 (exhibits)
San Diego, CA
Contact:
Infocom '83
PO Box 639
Silver Spring, MD 20901
(301) 589-8142

April 22–24
"ME" Technology Fair
San Bernardino, CA
Contact:
California State College,
San Bernardino
5500 State College Pkwy.,
San Bernardino, CA 92407

May 13–15
Applefest Boston
Boston, MA
Contact:
National Computer Shows
822 Boylston St.
Chestnut Hill, MA 02167
(617) 739-2000

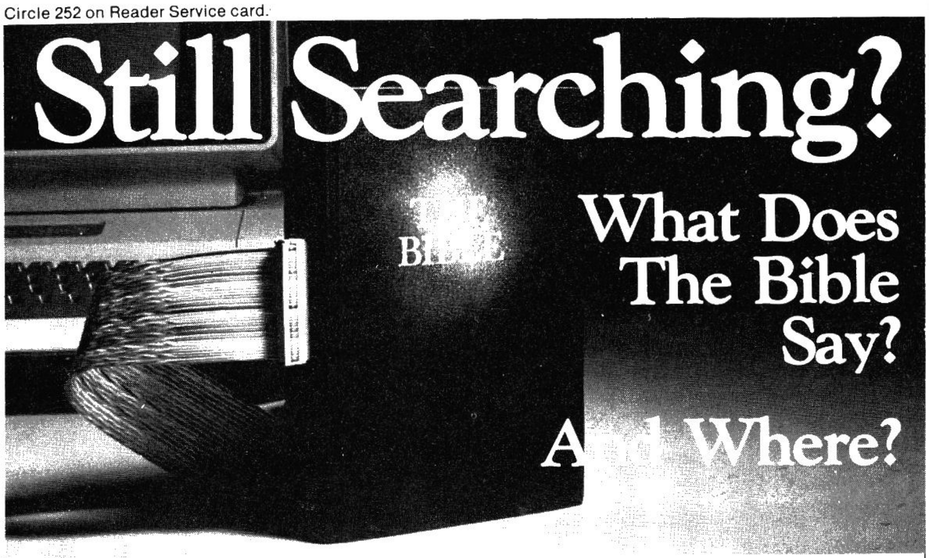
May 16–19
National Computer Conference
(exhibits)
Anaheim, CA
Contact:
AFIPS
1815 N. Lynn St. #800
Arlington, VA 22209
(703) 558-3600

May 23–26
Data Base Week
San Jose, CA
Contact:
Robin Williams
IBM Research Division
K55-282
5600 Cottle Road
San Jose, CA 95193

May 25
Trends and Applications 1983
Gaithersburg, MD
Contact:
Trends and Applications
PO Box 639
Silver Spring, MD 20901
(301) 589-8142

June 6–8
National Education Computer Conference
Baltimore, MD
Contact:
NECC
PO Box 639
Silver Spring, MD 20901
(301) 589-8142

June 26–29
Design Automation Conference
Miami Beach, FL
Contact:
Dr. Charles E. Radke
IBM Corporation
Hopewell Junction, NY 12533
(914) 485-7763



inCider BOOK SHELF

NANOS CARDS



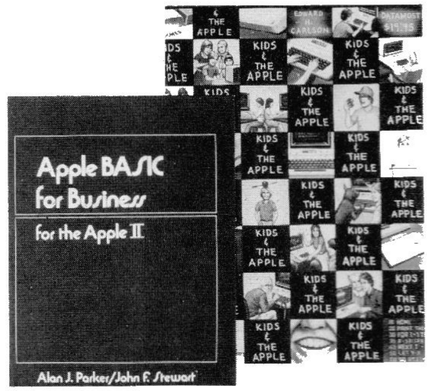
Nanos System REFERENCE CARDS FOR APPLE II, APPLE II PLUS, MODELS I, II, III AND COLOR COMPUTER

At last! No more flipping through the pages of the BASIC manual! No more working through the maze of machine language instructions! These cards completely summarize the BASIC and Assembler manuals! FEATURES INCLUDE:
memory map, eyeball graphics, math instructions, BASIC commands, store instructions, BASIC functions, load instructions, BASIC statements, move instructions, special keys, exchange instructions, PRINT USING examples, shift instructions, BASIC special characters, compare instructions, BASIC and assembler messages and codes, branch instructions, BASIC facts, data alteration instructions, reserved words, I/O instructions, ROM routines, complete character chart with graphics and space-compression codes, hex-dec chart, control code cross-reference, assembler instructions, commands and operators, screen line layout, editor commands and subcommands, condition code easy access.
Plus—"magic graphics number—a mystery until you learn how to use it!
Designed as a fold-up, accordion-style card, fits in your pocket. Panels organized for optimum speed for reference.

Apple II and Apple II Plus: BASIC and 6502 FC1008	$4.95
Apple II and Apple II Plus: BASIC only FC1007	$3.95
Model I: BASIC and Assembler FC1002	$4.95
Model II: BASIC and Assembler FC1005	$5.95
Model II: Commands and Utilities FC1010	$3.95
Model III: BASIC and Assembler FC1005	$5.95
Color: BASIC and Extended FC1006	$4.95
Z-80: Microprocessor FC1011	$4.95
ZX80, ZX81 and Timex Sinclair 1000: FC1012	$5.95
Pocket Computer: BASIC FC1009	$2.95

THE APPLE II USER'S GUIDE—By Lon Poole, Martin McNiff, and Steven Cook. This guide is the key to unlocking the full power of your Apple II or Apple II Plus. Topics include: "Applesoft and Integer BASIC Programming"—especially how to make the best use of Apple's sound, color and graphics capabilities. "Machine Level Programming," "Hardware Features"—which covers the disk drive and printer, and "Advanced Programming"—describing high resolution graphics techniques and other advanced applications. Well organized and easy to use. BK1220 $16.95

NEW



APPLE BASIC FOR BUSINESS: for the Apple II—by Alan J. Parker and John Stewart. Unlike most introductory BASIC books, this book uses files extensively. It is written specifically for the Apple II microcomputer with DOS Version 3.2. All programs presented are compatible with DOS Version 3.3. With the emphasis on problem-solving, the focus of this book is the point at which problem elements meet language capabilities. BK1247 $15.95

ASSEMBLY LANGUAGE PROGRAMMING FOR THE APPLE II—by Robert Mottolz. This comprehensive, easy to understand introduction provides solid groundwork for getting started in assembly language programming on the Apple II.' Many subroutines written in assembly language are provided, and most explanations are shown with equivalent examples in BASIC. There's an excellent section on hexadecimal arithmetic included, as well as appendices for further study. BK1249 $12.95

APPLE MACHINE LANGUAGE—by Don Inman and Kurt Inman. APPLE MACHINE LANGUAGE builds upon your previous knowledge of BASIC, and teaches you the machine language in small, easy, completely illustrated steps. Following this guide, you will be able to write machine language programs directly, using the Apple System Monitor. Each new program is thoroughly presented in functional blocks, with sketches of how each step will actually appear on the video screen. Soon you will be entering and executing your own machine language programs, with predictable results! BK1248 $14.95

MOSTLY BASIC: APPLICATIONS FOR YOUR APPLE II, BOOK 1—by Howard Berenbon. This book provides you with 28 ready-to-use, BASIC language programs which have been completely tested and debugged for use on your Apple II. Includes a telephone dialer, digital stop watch, spelling test, a house buying guide, a gas mileage calculator, and many others useful to businessmen, hobbyists, scientists, and computer enthusiasts. BK1251 $12.95

MOSTLY BASIC: APPLICATIONS FOR YOUR APPLE II, BOOK 2—by Howard Berenbon. A second gold mine of fascinating BASIC programs, including two dungeons that test your math and history abilities and another one that's strictly for fun, eleven household programs, a monthly savings plan and six more on money or investment, two that test your level of ESP, and more—32 in all! Excellent for beginning or advanced computerists. BK1252 $12.95

KIDS AND THE APPLE—by Edward H. Carlson. Whether you are a kid, a parent, or a teacher, this book is something unique. It starts with the bare bones introduction to programming, leads quickly to more interesting programs, and gives anyone who uses it a complete knowledge of Applesoft BASIC. Lively illustrations, notes to parents and teachers and questions for the reader are sprinkled throughout the book. While this guide is aimed at 8–16 year olds, adults will find it equally attractive as a beginning book for use the Apple personal computer! BK1253 $19.95



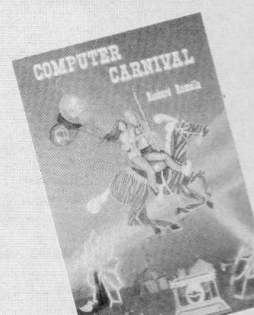
SOME COMMON BASIC PROGRAMS, APPLE II EDITION—By Lon Poole et al. A powerful collection of financial, statistical, home management and mathematics programs—76 in all. Each program is presented with BASIC source code, operating instructions and descriptions. If you're a beginning programmer you can learn from this book what well designed and documented programs look like. BK1232 $14.99

THE CUSTOM APPLE AND OTHER MYSTERIES—by Winifred Hofacker and Ekkehard Floegel. This is *the* guide to customizing Apple software and hardware, published by the folks at IJG. It contains such hands-on information such as: data acquisition and control applications, Programming the 6522 Internal timer, Constructing the 6522 I/O board, An Eprom Burner for the Apple Computer, An Eprom/RAM board, The Apple Slot Repeater, and much, much more. BK1246 $24.95.

FOR YOUR OTHER SYSTEM

COMPUTER CARNIVAL—by Richard Ramella. Your child can become a crackerjack computerist with the sixty TRS-80 Level II programs in COMPUTER CARNIVAL. This large-type, spiral bound book for beginners is a veritable funhouse of games, graphics, quizzes and puzzles. Written by *80 Micro* columnist Richard Ramella, the programs are challenging enough to ensure continued learning, yet short enough to provide your child with the immediate delight and reward of mastering basic computing skills. And for even greater enjoyment, get the CARNIVAL COMPANION, a 30-minute cassette containing all the programs in the book. Eliminates tiresome typing and lets your child spend more time enjoying the programs. BK7389 $16.97 CC7389 Book and Cassette $24.97

THE SELECTRIC INTERFACE—by George Young. You need the quality print that a daisy wheel printer provides but the thought of buying one makes your wallet wilt. SELECTRIC™ INTERFACE, a step-by-step guide to interfacing an IBM Selectric I/O Writer to your microcomputer, will give you that quality at a fraction of the price. George Young, co-author of *Kilobaud Microcomputing* magazine's popular "Kilobaud Klassroom" series, offers a low-cost alternative to buying a daisy wheel printer. SELECTRIC INTERFACE includes: step-by-step instructions, tips on purchasing a used Selectric, information on various Selectric models, including the 2740, 2980, and Dura 1041, driver software for Z80, 8080, and 6502 chips, tips on interfacing techniques. With SELECTRIC INTERFACE and some background in electronics, you can have a high-quality, low-cost, letter-quality printer. Petals not included. BK7388 (125 pages) $12.97



TEXTEDIT—A Complete Word Processing System in Kit Form—by Irwin Rappaport. TEXTEDIT is an inexpensive word processor that you can adapt to suit your differing needs—from form letters to lengthy texts. Written in TRS-80 Disk BASIC, the system consists of several modules, permitting the loading and use of only those portions needed. A disk is also available which provides the direct loading of the modules, however, the book is required for documentation. For Model I and III with TRSDOS CONVERT., one disk drive (2 disk drives or copy utility needed to transfer to system disk). Runs under TRSDOS 2.2/2.3. May not function under other systems. BK7387 $9.97 Disk DS7387 $19.97

For Toll Free Ordering Call 1-800-258-5473

*Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *inCider* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.50 for the first book, $1.00 each additional book for U.S. and foreign surface. $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

Index to Advertisers

*This advertiser prefers to be contacted directly. For further information from our advertisers, please use the Reader Service card.

INTRODUCING

SAVVY™



. . . the PERSONAL LANGUAGE™ system that mirrors your commands using your own words!

What SAVVY is —

- SAVVY is a miraculous new information handling system.
- SAVVY is an automatic database management system.
- SAVVY is a new level of machine intelligence.
- SAVVY, part hardware, part software, is the beginning of truly "Personal Computing".
- SAVVY comes with: General Ledger, Accounts Receivable, Accounts Payable, Payroll, Mailing List, Document Writer and Inventory Control.



Trademark: SAVVY, Robot Programmer: Excalibur Technologies Corporation.

Personal Language: SAVVY Marketing International.

What SAVVY does —

Through SAVVY, you and your computer talk to each other in your own natural, conversational English (or Spanish, or French, etc.).

It learns from you what you want done in your own personal language.

Once SAVVY learns your language it can create any file you wish. Input, output, additions, changes and deletions are arranged for you.

SAVVY's "Robot Programmer" has been trained to write 100% of the programs needed to manage your database information.

SAVVY runs CP/M™ and Apple DOS.

What YOU discover —

You'll discover that SAVVY recognizes your personal words, even if misspelled, or even if you use a phrase never used before!

SAVVY continues to grow through use to become better and better at understanding your commands.

Eventually, you will see SAVVY as a mirror to your own way of thinking and working. It is a re-definition of "user-friendly".

SAVVY, it's the first system that truly means "personal computing".

SAVVY is like no other system on earth.

SAVVY cost $950.

Seeing is believing. SAVVY is on display at selected computer retail locations. Call for the name of your nearest dealer.

CP/M is a trademark of Digital Research Corp.

Apple is a Trademark of Apple Computer Inc.

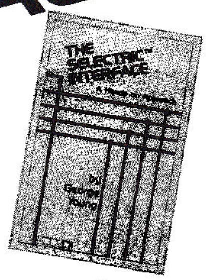
Circle 7 on Reader Service card.

1250 Oakmead Parkway Suite 210 Sunnyvale, CA 94086 (408) 773-1550

SAVVY

MARKETING INTERNATIONAL

THE SELECTRIC™ INTERFACE



Daisy wheel quality without daisy wheel expense.

You need the quality print that a daisy wheel printer provides but the thought of buying one makes your wallet wilt. *The Selectric™ Interface*, a step-by-step guide to interfacing an IBM Selectric I/O Writer to your microcomputer, will give you that quality at a fraction of the price. George Young, co-author of *Microcomputing* magazine's popular "Kilobaud Klassroom" series, offers a low-cost alternative to buying a daisy wheel printer.

The Selectric™ Interface includes:

- step-by-step instructions
- tips on purchasing a used Selectric™
- information on various Selectric™ models, including the 2740, 2980, and Dura 1041
- driver software for Z80, 8080, and 6502 chips
- tips on interfacing techniques

With *The Selectric Interface* and some background in electronics, you can have a high-quality, low-cost, letter-quality printer. Petals not included.

Credit card orders call TOLL-FREE 1-800-258-5473. Or mail your order with payment plus $1.50 shipping and handling to: Wayne Green Inc. Attn: Retail Book Sales, Peterborough, NH 03458.

Dealer inquiries invited.

ISBN 0-88006-051-4 128 pages **$12.97**

☐ Yes, I want **Selectric Interface** (BK7388). Enclosed is $12.97 per copy plus $1.50 for shipping and handling.

☐ MASTER ☐ VISA ☐ AMEX

Card # ______________ Expires ________

Signature ______________

Name ______________

Address ______________

City ______________

State and Zip ______________ 334B5S

All orders shipped UPS if complete street address is given.

Sales Opportunity

A few select territories available

Join the direct-sales force of a fast growing company involved in the microcomputer industry. Sales include our wide selection of books, six periodicals, and software. We are looking for a nonsmoker with a good sales background and a basic understanding of microcomputers. An excellent career awaits the person who wants to be "on his own."

It will be your responsibility to become familiar with all makes of computers and their sales outlets, amateur radio and electronics stores, book stores, and other outlets for our products in your territory. We will also add other lines as they become available.

Arrangements are being made to coordinate efforts with some other firms in the area. We offer an excellent growth potential with a highly motivated young corporation. If you (or someone you know)are interested in the opportunities available at Wayne Green Inc., please submit your background in detail, including sales experience, recent work history, knowledge of the microcomputing industry, educational background and references, to:

Sales Manager
Wayne Green Inc.
Peterborough, NH 03458

All replies will be acknowledged immediately and held in the strictest confidence.

WAYNE GREEN, INC.
(C-04) Pine Street
Peterborough, N.H.
03458

"AUTHOR AUTHOR!"

The call for authors is out!

Wayne Green Books announces a July 1, 1983 deadline for submitting manuscript proposals for the upcoming publication list. Ideas for book-length manuscripts about any microcomputer system or area of electronics will be considered. In addition to payment and royalties, we offer our distribution channels and the marketing support your book deserves.

Send proposals or requests for a copy of our Writer's Guide to:

Editor, **Wayne Green Books**
Peterborough, NH 03458.
Or call **toll-free 1-800-343-0728.**

Fudge It!

by Don Fudge

A renowned expert puts graphics shapes at your fingertips.

Block Shapes Made Easy

This column is about block shapes, table look-up for speed, simple hi-res page flipping as a flicker-curing technique, and turning anything on the hi-res screen into a block shape. (In the February column I supplied a program to turn anything on the hi-res screen into a vector shape.)

Last month I presented my TESTTB program, for block shape scanning and drawing, to type into the monitor and BSAVE (A2048,L140). It appears as Listing 4 in this month's column, and its applications are discussed below. More important than understanding the assembly language source codes behind this program is understanding what it is, what it does, when to use it, and what block shapes are.

Block Shapes

A *block shape* is a bit-mapped, or raster, shape. A block shape, in memory, is generally a long string of data. When one or more of these data sets are in memory, or stored on a disk or tape, you have a *block shape table*. In last month's column I covered a way to reconcile *which* hplot shape in an hplot shape table is being referenced. I use precisely the same method with block shapes, right down to the detail of POKE 7, SHAPENUMBER, to tell the program which shape is wanted.

When a block shape data set in memory becomes a block shape on the hi-res page, it no longer has data at sequential memory locations (unless it's only one line tall). See Figure 1. Only shape bytes on the same horizontal line (and adjacent) are sequential. (Check your white *Apple Reference Manual* for more on hi-res screen mapping.)

When one byte is directly beneath another on the hi-res screen, the bytes are *usually* 1024, or $400, address locations apart, as you can see from Figure 1. So, if you were to transfer the sequential data set of a block shape onto any part of hi-res page 1 (address locations $2000–$3FF8), leaving it sequential, you'd get no more than a series of weird, choppy lines, spaced well apart.

The goal is to have the block shape end up in the same visible shape as it was in the original creation. For this purpose use either the Applesoft HPOSN routine or the table look-up method covered here. The end result of either is that the problem of the mixed-up hi-res screen mapping is resolved and the sequential data bytes stored in memory are loaded non-sequentially into the appropriate hi-res locations.

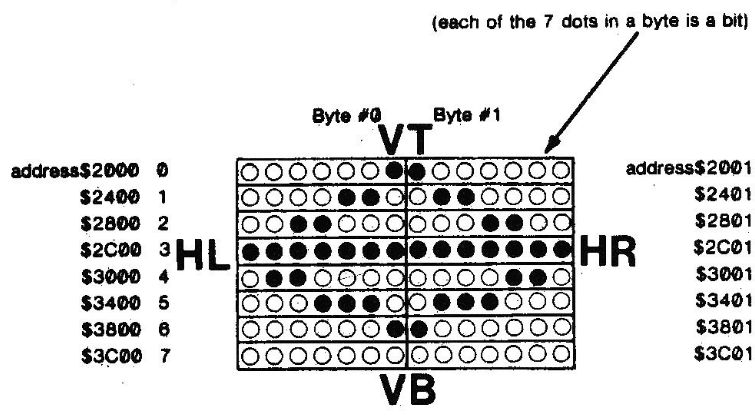


Figure 1. Block shape on a hi-res page.

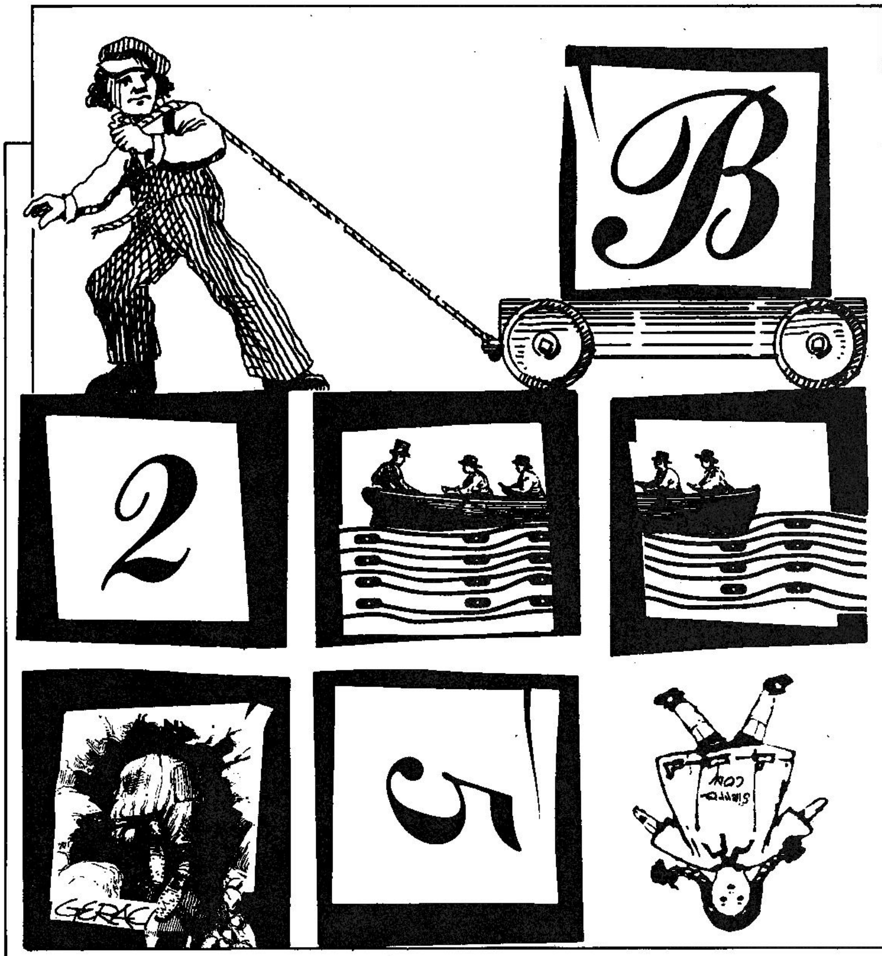
Let's say a block shape 16 bytes long resides at $900 to $90F. Call it shape 1 of a three-shape block shape table, named "flying saucer," stored at address $900. Shape 1 is one orientation of this flying saucer. Shapes 2 and 3 are two different orientations (or views, or rotations, or scale-ups) of the saucer.

In my style of block shape table, shape 1 starts at $900, shape 2 starts at $A00, and shape 3 starts at $B00. The routine that handles these shapes is the aforementioned TESTTB (Listing 4) and is stored at $800.

Okay, now consult Figure 1 again. If your block shape data needs to end up at the memory locations given, then obviously you'll be grabbing bytes from the flying saucer data table at $900–$90F and dispositioning them hither, thither, and yon. TESTTB makes sure the $900 table's bytes are appropriately placed onto the hi-res screen.

I've used HPOSN to get the sorting out done, in TESTTB. It's slower than table look-up, but it uses less memory, and memory's often at a premium when I'm at the keyboard. Notice that bytes are not "taken from" the shape table and placed onto the screen. Instead, bytes are

Address correspondence to Don Fudge at Avant-Garde Creations, PO Box 30160, Eugene, OR 97403.



copied from the table and then the data copied (such as $7F) is put into the appropriate hi-res screen address. The table doesn't change at all.

The shape in Figure 1 is 2 bytes wide by 8 bytes tall. Notice that only 7 of the 8 bits in each byte are shown. The eighth bit is a flag, not a screen bit, so it isn't displayed. Some microcomputer hi-res mapping allows the convenience of 8 visible bits per byte, but Apple opted not to. Also, remember that bytes are displayed backwards on the screen. Normally the bytes at $2000 and $2001 would be read as having values of 1 and 64. But $2000 contains 64 (or 192 if the high, color flag, bit is on) and $2001 contains 1 (or 129 if the high, color flag, bit is on).

In using block shapes a la Fudge, you need to make five specifications.
1) The shape number (1–23); POKE it into 7 (decimal memory location).
2) The VT (vertical top coordinate—the highest Y coordinate in the shape); POKE it into 252.
3) The VB (vertical bottom coordinate—the lowest Y coordinate); POKE it into 253.
4) The HR (horizontal right coordinate—the rightmost X coordinate); POKE it into 254.
5) The HL (horizontal left coordinate—the leftmost X coordinate); POKE it into 255.

Keep in mind that Y coordinates refer to numbers from 0 to 191, as is normal for Apple hi-res, but with block shapes X coordinates refer to *bytes*. There are 40 horizontal bytes, labeled #0 to #39 on the Apple's screen. So, no X coordinate going into decimal locations 254 (for HR) or 255 (for HL) may exceed 39.

Now, the shape in Figure 1 is in the upper left corner of the screen. That means its coordinates are as follows: VT = 0, VB = 7, HR = 1, and HL = 0. Notice what would happen if you tried to get the shape's dimensions by subtracting HL from HR for width: the shape would be 7 by 1. That won't do. The shape is really 8 by 2. However, it's convenient to let VB–VT = height and HR–HL = width, so do it. From now on, *refer* to width as 1 and height as 7. Thus, you can use 160 as VT and 167 as VB and know that you've merely added height and VT to get VB. But, you'll *also* know that in using coordinates 160 through 167 you'll be using eight different coordinate lines, not seven. The same goes for width. You may add 1 to an HL of 13 to get 14 (as the HR) and realize that, even though you *say* 1 wide, you'll be using bytes 13 *and* 14, so it's really 2 bytes wide. Inclusivity is the key here.

So, no matter where the block shape was originally drawn, you're permitted, when using TESTTB, to DRAW, or XDRAW a block shape anywhere on the screen. You must only make sure VB<192, HR<40, VB–VT = height, and HR–HL = width (not *true* height/width, but height/width *to use in figuring*). Also, don't forget which shape number goes with which shape, and remember to jot down height and width for all shapes created and save this information.

Scanning

You've seen how data in a table ends up correctly dispositioned on the screen. But you may well ask how screen bytes end up correctly dispositioned in a table in the first place. Thought you'd never ask!

In scanning or drawing a block shape's bytes, I opted to begin at the lower right and move to the left until running out of bytes, then to move up a line, start at the far right and move to the left again. This continues until the upper left byte of the block shape is reached. I could as easily have gone from left to right and top to bottom.

Consult Figure 1. If you were scanning the shape block from bottom right to upper left and then putting the byte copies sequentially into $900–$90F, the data from $3C01 would be copied into $900, the data from $3C00 would go to $901, and the process would continue until the $2000 data copy ended up in $90F. DRAWing on the screen would entail the reverse of the scanning process—you'd put table data copies into screen locations.

In case you don't quite see what the word *scan* has to do with all this, I'll explain. When you get an image onto the hi-res screen you have just

turned bits on in various screen byte addresses (unless you're drawing on a white, or on, screen, in which case XDRAWing results in a black figure and the turning *off* of bits in bytes). If you decide to have a permanent bit-mapped image of a certain section of the screen (a block shape), you tell the Apple what part of the screen you want as a block shape, using game paddles to *paddle-define* the block shape desired. Copy *those* data numbers into a table (from $900 up to as high as $1FFF) and then BSAVE this binary data file onto disk as a *disk-based shape table*. The process of going from byte to byte and copying data from hi-res locations to table locations is called *scanning*. To scan with TESTTB you CALL 2048 after giving the shape number in location 7 and coordinates VT, VB, HR, and HL in 252–255. To XDRAW with TESTTB you CALL 2116 after giving the shape number and coordinates as above. To DRAW with TESTTB you POKE 2153,234 and POKE 2154,234 first. To restore XDRAW you POKE 2153,81 and POKE 2154,38 first.

The difference between DRAW and XDRAW is EOR ($26), Y. XDRAW has this command; in the DRAW routine POKEs put NOPs (no operation) in place of it, and the XDRAW restorer changes NOP NOP to EOR ($26), Y.

The EOR command makes bytes loaded to the screen result in a shape opposite in color to the background. Two EOR-XDRAWs in a row allow you to leave the screen exactly as found. In EOR, 1 and 1 is 0, 1 and 0 is 1, and 0 and 0 is 0. Therefore, resultant shape bits are "on" (1) only if the shape bit is opposite in setting from the background bit being "landed upon." XDRAW is usually used to make sure no permanent harm comes to the background. It can be used for erasing more easily than DRAWing with HCOLOR = 0 (black).

If a test for collision is needed, then you'll want to AND the shape being drawn with the background, as a test. If the AND result isn't 0, there is a collision of the shape with the background, *which may be a shape already on the screen*. AND is an instruction in which 1 and 1 is 1, 0 and

Listing 1. SCANA.

```
0  ONERR  GOTO 63990
1  PRINT  CHR$ (4);"BLOADTESTTB": GOSUB 2500: GOTO 600
2  HOME : INPUT "SHAPE TABLE NAME: ";STN$: IF  LEN (STN$) = 0 THEN 600
3  INPUT "DEC. STARTING ADDR. OF TABLE (MAKE SURE YOU BLOAD IT ABOVE 32768
    ---EVEN IF YOUR VECTOR SHAPE TABLE IS LOWER THAN THIS,  HAVE IT BLOAD
     ABOVE 32768 NOW BY TYPING A # AT LEAST THAT BIG NOW):";AD
4 D$ =  CHR$ (4): PRINT D$"BLOAD";STN$;",A";AD
5  HOME : VTAB 21: INPUT "SHAPE #: ";SHN: POKE 7,SHN
10  POKE 232,AD - ( INT (AD / 256) * 256): POKE 233, INT (AD / 256)
15  POKE  - 16304,0: POKE  - 16297,0
18 VS = 1:BS = 0
20  INPUT "SHAPE X COORDINATE: ";XX: INPUT "SHAPE Y COORDINATE: ";YY
42  HCOLOR= H: ROT= R: SCALE= S
43  DRAW SHN AT XX,YY: HOME : VTAB 21: INPUT "DO YOU WANT ANOTHER SHAPE? (
    Y/N):";QW$: IF  LEN (QW$) = 0 THEN 43
44  IF  ASC (QW$) <  > 89 THEN  HOME : VTAB 21: GOSUB 63000: GOTO 600
45  GOTO 5
47  POKE  - 16303,0: POKE  - 16298,0: HOME : VTAB 1: PRINT "USE THE PADDLE
    S TO MOVE THE DOT TO THE  UPPER LEFT RECTANGLE POINT. HIT PDL 0   BUT
    TON. THEN MOVE THE DOT TO THE LOWER  RIGHT RECTANGLE POINT. HIT PDL 1
     BUTTON.": GOSUB 63000
48  POKE 232,248: POKE 233,8: SCALE= 1: ROT= 64
49  POKE  - 16304,0: POKE  - 16297,0
50  HOME :P1 =  PDL (1): IF P1 > 159 THEN 50
55 P0 =  PDL (0): XDRAW 1 AT P0,P1:X% = P0:Y% = P1
60 P1 =  PDL (1): IF P1 > 159 THEN 60
65  FOR QW = 1 TO 200: NEXT : HOME : VTAB 21: PRINT "X: "P0: PRINT "Y: "P1

70 P0 =  PDL (0): XDRAW 1 AT X%,Y%: XDRAW 1 AT P0,P1:X% = P0:Y% = P1
80 B0 =  PEEK ( - 16287): IF B0 > 127 AND FL = 0 THEN FL = 1: GOTO 100
85 B1 =  PEEK ( - 16286): IF B1 > 127 AND SG = 0 THEN SG = 1: GOTO 110
90  GOTO 60
100 VT = P1:HL =  INT (P0 / 7): PRINT  CHR$ (7): IF SG = 1 THEN 120
105  GOTO 60
110 VB = P1:HR =  INT (P0 / 7): PRINT  CHR$ (7): IF FL = 1 THEN 120
115  GOTO 60
120  HOME : VTAB 21: PRINT "HOR.--FROM:"HL" TO "HR"---WIDTH:"(HR - HL)
125  XDRAW 1 AT P0,P1
130  PRINT "VER.--FROM:"VT" TO "VB"---HEIGHT:"(VB - VT): VTAB 23: PRINT "J
     OT THIS DOWN! (HIT ANY KEY TO CONT.)": GOSUB 63010
150  POKE 252,VT: POKE 253,VB: POKE 254,HRIGHT: POKE 255,HLEFT
155  HCOLOR= 3
160  HPLOT 7 * HRIGHT + 7,VT TO 7 * HRIGHT + 7,VB TO 7 * HLEFT,VB TO 7 * H
     LEFT,VT TO 7 * HRIGHT + 7,VT
170  IF ZQ = 1 THEN  RETURN
180  PRINT : INPUT "IS THE RECTANGLE DONE O.K.? (Y/N):";AN$: IF  LEN (AN$)
      = 0 THEN 180
185  IF  ASC (AN$) = 78 THEN SG = 0: HCOLOR= 0:FL = 0:ZQ = 1: GOSUB 160:ZQ
      = 0: HCOLOR= 3: GOTO 50
186  INPUT "SHALL WE SCAN THIS SHAPE & PDL-DEFINE   OTHERS IN THIS TABLE T
     OO, TO BE USED AS ENTRIES FOR BLOCK TABLE WE'RE ON?(Y/N)";QW$: IF  LEN
      (QW$) = 0 THEN 186
187  IF  ASC (QW$) <  > 89 THEN 191
188  INPUT "BLOCK SHAPE #:";SHN
189  POKE 7,SHN:ZQ = 1: HCOLOR= 0: GOSUB 160: HCOLOR= 3:ZQ = 0: CALL 2048:
      IF BS = 1 THEN BS = 0:VS = 0: GOSUB 1000: GOTO 406
190 BS = 0:VS = 0: GOSUB 1000: GOTO 5
191  GOTO 600
204  HOME : VTAB 1
205  INPUT "GIVE THE SHAPE # YOU WANT ALL THE        SCANNED BYTES' DATA TO
      GO INTO; REMEMBERTHAT LONG SHAPES CAN'T START TOO NEAR   SHAPE # 23.
      THE TABLE WE'RE CREATING    WILL BE LESS THAN 24 SHAPES LONG & LESS
     THAN 5888 BYTES (1-23):";SHN
206  POKE 7,SHN: POKE  - 16304,0: POKE  - 16297,0
210 ZQ = 1: HCOLOR= 0: GOSUB 160: HCOLOR= 3: CALL 2048
300 D$ =  CHR$ (4)
301  VTAB 21
302  INPUT "FILE NAME: ";N$: IF  LEN (N$) = 0 THEN 302
303  INPUT "DID YOU GET IT RIGHT? (Y/N):";Z$: IF  LEN (Z$) = 0 THEN 302
304  IF  ASC (Z$) <  > 89 THEN 302
307  TEXT : VTAB 1: HOME : GOSUB 5000
308 LL = ((VB - VT) + 1) * ((HR - HL) + 1) + ((SHN - 1) * 256)
309 PL = 1 +  INT ((((VB - VT) + 1) * ((HR - HL) + 1)) / 256)
310  PRINT D$"BSAVE";N$;",A2304,L";LL
312  PRINT "THIS SHAPE AND ALL THE SHAPES THAT CAME BEFORE IT TOOK UP "LL"
      BYTES.": PRINT "IT'S # "SHN" SO GIVE IT "PL" PLACES IN": PRINT "THE
     FILE. (HIT ANY KEY TO CONTINUE):": GOSUB 63010
400  GOTO 600
402  HOME : VTAB 1: INPUT "SHAPE TABLE NAME: ";STN$: IF  LEN (STN$) = 0 THEN
     600
403  INPUT "DID YOU GET IT RIGHT? (Y/N):";QW$: IF  LEN (QW$) = 0 THEN 402
404  IF  ASC (QW$) <  > 89 THEN 402
405 D$ =  CHR$ (4): PRINT D$"BLOAD";STN$
406  INPUT "SHAPE #: ";SHN: POKE 7,SHN
407  POKE  - 16304,0: POKE  - 16297,0: HOME : VTAB 21
408  INPUT "VTOP: ";VT: INPUT "VBOT: ";VBOT: INPUT "HRIGHT: ";HR: INPUT "H
     LEFT: ";HL
410  POKE 252,VT: POKE 253,VBOT: POKE 254,HRIGHT: POKE 255,HLEFT
500  CALL 2116
510 BS = 1:VS = 0
525  HOME
530  VTAB 21: INPUT "DO YOU WANT ANOTHER SHAPE? (Y/N):";QW$: IF  LEN (QW$)
      = 0 THEN 530
540  IF  ASC (QW$) <  > 89 THEN  HOME : GOTO 550
545  GOTO 406
550  VTAB 21: GOSUB 63000
600  POKE  - 16303,0: POKE  - 16298,0: HOME : VTAB 1: INVERSE : HTAB 18: PRINT
     "MENU:": NORMAL
601 SG = 0:FL = 0:ZQ = 0
602  SCALE= S: HCOLOR= H: ROT= R
603  PRINT "(HIT ESC TO QUIT)": PRINT
605  PRINT "(0)ABORT SCREEN---START OVER": PRINT
610  PRINT "(1)LOAD VECTOR SHAPE TABLE": PRINT
```

Listing continued.

Listing continued.

```
620  PRINT "(2)CHOOSE A VECTOR SHAPE COLOR": PRINT
630  PRINT "(3)CHOOSE A VECTOR SHAPE SCALE": PRINT
633  PRINT "(4)CHOOSE A VECTOR SHAPE ROTATION": PRINT
636  PRINT "(5)CHOOSE A V. SHAPE BACKGROUND COLOR": PRINT
640  PRINT "(6)LOAD IN A BLOCK SHAPE": PRINT
650  PRINT "(7)DEFINE BLOCK SHAPE WITH PADDLES": PRINT
660  PRINT "(8)VIEW SCREEN": PRINT
670  PRINT "(9)SAVE PADDLE-DEFINED BLOCK AS FILE": PRINT
690  FLASH : PRINT "(CHOOSE 0-9):";: NORMAL : GET A$: PRINT  CHR$ (13)
691  REM
692  IF  ASC (A$) = 27 THEN  TEXT : HOME : END
700  IF  LEN (A$) = 0 THEN 690
710  IF  VAL (A$) < 0 OR  VAL (A$) > 9 THEN 690
719  IF A$ = "0" THEN 912
720  ON  VAL (A$) GOTO 2,900,904,908,916,402,47,920,204
721  GOTO 600
900  HOME : VTAB 1: INPUT "SHAPE COLOR? (1-7):";H: IF H > 7 OR H < 0 THEN
     900
902  GOTO 600
904  HOME : VTAB 1: INPUT "SCALE? (1-255):";S: IF S > 255 OR S < 1 THEN 90
     4
906  GOTO 600
908  HOME : VTAB 1: INPUT "ROTATION? (1-255):";R: IF R > 255 OR S < 1 THEN
     908
910  GOTO 600
912  INPUT "SURE YOU WANT TO ABORT SCREEN? (Y/N):";QW$: IF  LEN (QW$) = 0 THEN
     912
913  IF  ASC (QW$) <  > 89 THEN 600
914  HGR : GOTO 600
916  INPUT "SURE YOU WANT TO DO A BACKGROUND COLOR? THIS WILL ERASE ANY SH
     APE! (Y/N):";QW$: IF  LEN (QW$) = 0 THEN 916
917  IF  ASC (QW$) <  > 89 THEN 600
918  HOME : VTAB 1: INPUT "BACKGROUND COLOR (THIS ERASES SHAPES!)  (1-7):"
     ;BC: IF BC > 7 OR BC < 0 THEN 918
919  POKE  - 16304,0: POKE  - 16297,0: HCOLOR= BC: HPLOT 0,0: CALL 62454: VTAB
     21: GOSUB 63000: GOTO 600
920  POKE  - 16304,0: POKE  - 16297,0: VTAB 21: GOSUB 63000: GOTO 600
1000  HOME : VTAB 21: INPUT "DO YOU WANT TO ERASE SCREEN BEFORE      SCANN
     ING NEXT SHAPE (NECESSARY WHEN YOU ARE DOING ANIMATION SEQUENCES) (Y/
     N):";QW$: IF  LEN (QW$) = 0 THEN 1000
1010  IF  ASC (QW$) <  > 89 THEN  RETURN
1020  CALL 62450: RETURN
2500  POKE 2296,1: POKE 2297,0: POKE 2298,4: POKE 2299,0: POKE 2300,4: POKE
     2301,0
2510  SCALE= 1:S = 1: ROT= 64:R = 64: HCOLOR= 3:H = 3: POKE  - 16301,0
2511  POKE  - 16303,0: POKE  - 16298,0: INVERSE : PRINT "IF YOU ENTERED TH
     IS PROGRAM WITH SOME-  THING ON THE HI-RES SCREEN YOU WANTED TOSAVE,
     HIT THE SPACE BAR NOW---          OTHERWISE HIT ANY KEY EXCEPT THE SP
     ACE  BAR.": NORMAL
2512 PK =  PEEK ( - 16384): IF PK > 127 THEN  POKE  - 16368,0: GOTO 2514
2513  GOTO 2512
2514  IF PK = 160 THEN 2520
2515  HGR
2520  RETURN
5000  PRINT "WOULD YOU LIKE TO GIVE THE # OF SHAPES  IN YOUR TABLE & HAVE
     THIS PROGRAM GIVE  EACH SHAPE 256 BYTES OF ROOM, OR WOULD  YOU RATHER
      LET THE LAST SHAPE # YOU      ENTERED BE THE LAST SHAPE IN THE TABLE
      YOU'LL SAVE NOW?": PRINT
5010  PRINT "(THE MAXIMUM SHAPES IS 23 AND THE MAXI- MUM BYTES IS 5888. IF
      ANY OF YOUR SHAPESWERE OVER 256 BYTES LONG, THEY COUNT AS 2 SHAPES.
     OVER 512 COUNTS AS 3, ETC.)": PRINT
5020  PRINT "(1)GIVE # OF SHAPES": PRINT "(2)LAST SHAPE # GIVEN CAN BE LAS
     T ONE IN   THE TABLE": PRINT : INPUT "(1-2):";QW$: IF  LEN (QW$) = 0 THEN
     5020
5030  IF  ASC (QW$) <  > 49 THEN  RETURN
5040 .PRINT : INPUT "# OF SHAPES:";NS: IF NS < 1 OR NS > 23 THEN 5040
5050 LL = 256 * NS: POP : GOTO 309
63000  PRINT "                                   (HIT ANY KEY TO CONT
     INUE):              "
63010 PK =  PEEK ( - 16384): IF PK > 127 THEN  POKE  - 16368,0: RETURN
63020  GOTO 63010
63990  PRINT  CHR$ (7): POKE 216,0
63991 PP =  PEEK (222): IF PP = 254 THEN  RESUME
63994  POKE  - 16303,0: POKE  - 16298,0
63995  PRINT "YOUR ERROR IS CODE #:"PP: GOSUB 63000: CALL 54915: GOTO 600
```

```
0  REM  PREPARATION FOR ANIMATION
10  HGR : POKE  - 16302,0:Q = 0: HGR2 : REM :INITIALIZE BOTH PAGES TO FULL
      SCREEN
20  HCOLOR= 3: REM  WHITE
30  POKE 230,64: POKE  - 16300,0: REM  DRAW ON 2; DISPLAY 1
40  HPLOT Q,0 TO Q,191: REM  DRAW LINE ON 2
50  POKE 230,32: POKE  - 16299,0: REM  DRAW ON 1; DISPLAY 2
60  HPLOT Q + 1,0 TO Q + 1,191: REM MOVE 1 STEP AND DRAW PAGE 1 LINE
200  REM  PREPARATION OVER; LOOP BEGINS
210  POKE 230,64: POKE  - 16300,0: REM DRAW ON 2; DISPLAY 1
220  HCOLOR= 0: REM   BLACK FOR   ERASING
230  HPLOT Q,0 TO Q,191: REM.       ERASE
240  HCOLOR= 3: REM  WHITE
250 Q = Q + 2: REM  MOVE 2 STEPS, ONE STEP GETS YOU OFF ERASE  POSITION &
     ONTO OPPOSITE   PAGE'S DRAWN LINE, 2ND STEP   GETS YOU TO NEXT DRAWIN
     G   POSITION
260  HPLOT Q,0 TO Q,191: REM       DRAW LINE ON 2
270  POKE 230,32: POKE  - 16299,0: REM DRAW ON 1; DISPLAY 2
280  HCOLOR= 0: REM   BLACK FOR   ERASING
290  HPLOT Q - 1,0 TO Q - 1,191: REM ERASE ONE BEHIND OPPOSITE   PAGE'S LI
     NE & DRAW ONE AHEAD   OF OPPOSITE PAGE'S LINE
300  HCOLOR= 3: REM  WHITE
310  HPLOT Q + 1,0 TO Q + 1,191: REM  DRAW ONE AHEAD OF OPPOSITE PAGE'S LI
     NE
320  IF Q > 276 THEN  END : REM  DON'T GO OFF THE EDGE OF THE WORLD
330  GOTO 210: REM   LOOP BACK &  CONTINUE ERASE/DRAW CYCLES
```

Listing 2. Two-Page Animation.

0 is 0, and 1 and 0 is 0, so only 2 *on* bits colliding creates a non-zero result. If you're using high-bit-on colors (blue, orange, black2, white2) then you'll want to AND the background with #$7F before doing the AND test.

Animation

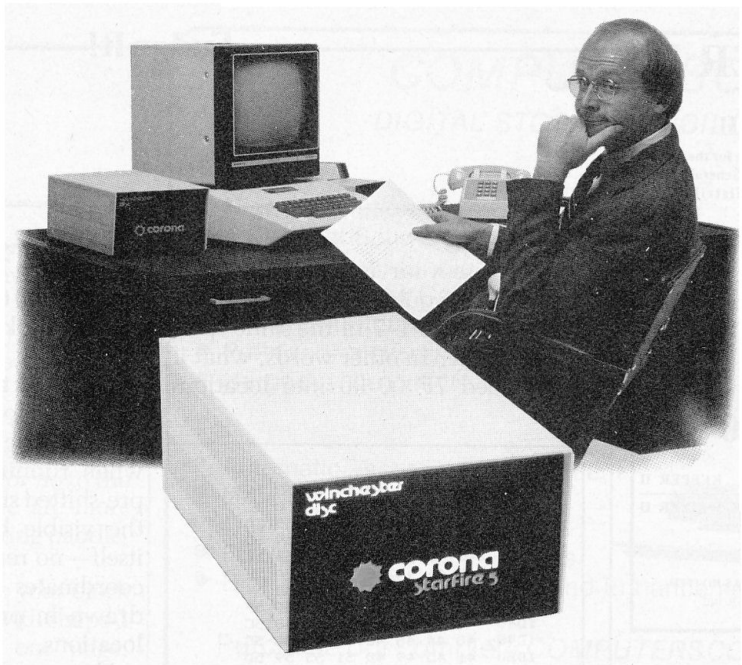
Animation with block shapes is much more complex than vector shape animation. Hi-Res Secrets (from Avant-Garde Creations) has lots of one-page and two-page flipping animation routines for block shape movement. The core of the problem, the aspect of block shapes that creates the extra complexity, is the fact that only 40 horizontal coordinates are possible, and each one is a full 7 bits away from its neighbors. This means that if you're using only one block shape, your animation movements must be 7 dots wide (or 14 or 21, and so forth).

Even if 7-dot movements are okay, you'll get color-flip because color consistency is based on doing all drawing at odd (only) or even (only) X coordinates, not odd/even mixtures. So, only white shapes may be used with 7-dot movements (also known as 1-byte movements).

In practice, animation is nearly always in steps of 1 (white only), 2 (color-safe), or 4 (also color-safe) bits sideways. But to effect this we use what some call *pre-shifted shapes*, and what I call *shape sequences* in Hi-Res Secrets. Seven shapes are the usual sequence. See Table 1.

You'll notice that a one-dash shape is to be moved from left to right. The step factor (dots/movement) is 2, which is color-safe. However, since the shape is an all-white dash (with color, every other bit is *off*, but with the dash all dash-bits are *on*) this color-safeness is of no consequence.

The far-left numbers are shape numbers. Notice that shape 2 is the same as shape 1, except that all bits have been shifted right (with an ASL or ROL, since screen bytes are displayed backwards). By the way, each shape in a block shape sequence must have the same width and screen coordinates, and yet the visible shape within them is in a different position



To Do It Right.

To make a Winchester disk for just Apple®II.

That's how we set out to design our hard disk for the Apple®II. To understand the needs of serious users and programmers, and to correct the errors of our predecessors.

You want user-friendliness. So we wrote hard disk versions of Apple DOS, CP/M® and Pascal that are highly user-friendly and loaded with useful features.

- ☐ Auto-boot hard disk
- ☐ Menu-driven utilities
- ☐ Single-keystroke program execution in DOS
- ☐ Disk search with wild cards, and many more

You want flexibility. So we made every feature variable.

- ☐ From 1 to 16 operating systems on each disk
- ☐ Operating system spaces grow as needed
- ☐ Slot independence
- ☐ Variable size volumes (to 400K in DOS)
- ☐ Mountable and unmountable volumes (even in CP/M)
- ☐ Custom DOS allowed

You want reliability. We use the leading drive. And Corona's unique data protection technology.

- ☐ DataGuard™ 32-bit error correction code
- ☐ FailSafe™ read-after-write and automatic bad-sector reallocation
- ☐ 2-level impact-protection packaging

You want low-cost backup. So we wrote backup utilities that make floppy backup convenient.

- ☐ File compacting to reduce the number of floppies
- ☐ Volume selective backup in all operating systems
- ☐ Automatic diskette sequencing to ease floppy handling

You want compatibility. We maximize compatibility with existing software and peripheral cards.

- ☐ 9K interface card leaves main memory untouched
- ☐ Interrupts are allowed
- ☐ Boot protected floppies from slot 6 without removing hard disk
- ☐ Automatic slot/drive to hard disk remapping

And you want support. We do that right too.

- ☐ Hardware depot service in every region
- ☐ Software theft-protection on the hard disk

Compare the features that matter to you. And visit our local dealer or distributor for a demonstration. You'll see the difference specialization makes.

Corona Starfire™— The Winchester Disk for Apple II

~~$2495~~ / 5 MB ~~$2995~~ / 10 MB

(All software included. Pascal not needed for CP/M.)

Now $2195 Now $2695

corona™

The Third Generation Microcomputer Company

Corona Data Systems • 31324 Via Colinas, Section 110, Westlake Village, CA 91361 • 213-706-1505

Circle 175 on Reader Service card.

for each one.

Now use your imagination for a bit. What would it look like if we "drew" shapes 1–7 in the same spot, sequentially? In other words, what if we loaded 7F 00 00 into locations $2000–2002, and then loaded 7C 03 00 and then 70 0F 00, and so on, until we got to 00 60 1F? What if the whole process took a mere ½ second? It would look like a dash sliding sideways, even though our shape drawing is *all* into locations $2000 through $2002. The "movement" effected when running a shape sequence of pre-shifted shapes is the movement of the visible bits of the block shape itself—no real change of shape block coordinates occurs. All shapes are drawn in precisely the same screen locations.

However, once the sequence is exhausted, the HL and HR must be updated the same number of units as there are dots in each visible step-movement. In other words, the dash moved 2 dots per movement step within the block shape, therefore the HL and HR coordinates of the block shape must be incremented 2 bytes after every shape sequence is completed. This means that shapes 1–7 are "drawn" sequentially on top of each other, always using the same VT, VB, HR and HL, but once shape 7 is drawn it's time to say HL = HL + STEPSIZE and HR = HR + STEPSIZE in order to continue the sideways dash movement. *Vertical* animation, on the other hand, takes no pre-shifted shapes to accomplish. In either case, page-flipping is often part of the animation process. We'll get to that, as Starbuck would say, "in a few centons."

```
1D00- 40 44 48 4C 50 54 58 5C
1D08- 40 44 48 4C 50 54 58 5C
1D10- 41 45 49 4D 51 55 59 5D
1D18- 41 45 49 4D 51 55 59 5D
1D20- 42 46 4A 4E 52 56 5A 5E
1D28- 42 46 4A 4E 52 56 5A 5E
1D30- 43 47 4B 4F 53 57 5B 5F
1D38- 43 47 4B 4F 53 57 5B 5F
1D40- 40 44 48 4C 50 54 58 5C
1D48- 40 44 48 4C 50 54 58 5C
1D50- 41 45 49 4D 51 55 59 5D
1D58- 41 45 49 4D 51 55 59 5D
1D60- 42 46 4A 4E 52 56 5A 5E
1D68- 42 46 4A 4E 52 56 5A 5E
1D70- 43 47 4B 4F 53 57 5B 5F
1D78- 43 47 4B 4F 53 57 5B 5F
1D80- 40 44 48 4C 50 54 58 5C
1D88- 40 44 48 4C 50 54 58 5C
1D90- 41 45 49 4D 51 55 59 5D
1D98- 41 45 49 4D 51 55 59 5D
1DA0- 42 46 4A 4E 52 56 5A 5E
1DA8- 42 46 4A 4E 52 56 5A 5E
1DB0- 43 47 4B 4F 53 57 5B 5F
1DB8- 43 47 4B 4F 53 57 5B 5F
1DC0- FF FF 00 00 FF FF 00 00
1DC8- FF FF 00 00 FF FF 00 00
1DD0- FF FF 00 00 FF FF 00 00
1DD8- FF FF 00 00 FF FF 00 00
1DE0- FF FF 00 00 FF FF 00 00
1DE8- FF FF 00 00 FF FF 00 00
1DF0- FF FF 00 00 FF FF 00 00
1DF8- FF FF 00 00 FF FF 00 00
1E00- 00 00 00 00 00 00 00 00
1E08- 80 80 80 80 80 80 80 80
1E10- 00 00 00 00 00 00 00 00
1E18- 80 80 80 80 80 80 80 80
1E20- 00 00 00 00 00 00 00 00
1E28- 80 80 80 80 80 80 80 80
1E30- 00 00 00 00 00 00 00 00
1E38- 80 80 80 80 80 80 80 80
1E40- 28 28 28 28 28 28 28 28
1E48- A8 A8 A8 A8 A8 A8 A8 A8
1E50- 28 28 28 28 28 28 28 28
1E58- A8 A8 A8 A8 A8 A8 A8 A8
1E60- 28 28 28 28 28 28 28 28
1E68- A8 A8 A8 A8 A8 A8 A8 A8
1E70- 28 28 28 28 28 28 28 28
1E78- A8 A8 A8 A8 A8 A8 A8 A8
1E80- 50 50 50 50 50 50 50 50
1E88- D0 D0 D0 D0 D0 D0 D0 D0
1E90- 50 50 50 50 50 50 50 50
1E98- D0 D0 D0 D0 D0 D0 D0 D0
1EA0- 50 50 50 50 50 50 50 50
1EA8- D0 D0 D0 D0 D0 D0 D0 D0
1EB0- 50 50 50 50 50 50 50 50
1EB8- D0 D0 D0 D0 D0 D0 D0 D0
1EC0- 20 24 28 2C 30 34 38 3C
1EC8- 20 24 28 2C 30 34 38 3C
1ED0- 21 25 29 2D 31 35 39 3D
1ED8- 21 25 29 2D 31 35 39 3D
1EE0- 22 26 2A 2E 32 36 3A 3E
1EE8- 22 26 2A 2E 32 36 3A 3E
1EF0- 23 27 2B 2F 33 37 3B 3F
1EF8- 23 27 2B 2F 33 37 3B 3F
1F00- 20 24 28 2C 30 34 38 3C
1F08- 20 24 28 2C 30 34 38 3C
1F10- 21 25 29 2D 31 35 39 3D
1F18- 21 25 29 2D 31 35 39 3D
1F20- 22 26 2A 2E 32 36 3A 3E
1F28- 22 26 2A 2E 32 36 3A 3E
1F30- 23 27 2B 2F 33 37 3B 3F
1F38- 23 27 2B 2F 33 37 3B 3F
1F40- 20 24 28 2C 30 34 38 3C
1F48- 20 24 28 2C 30 34 38 3C
1F50- 21 25 29 2D 31 35 39 3D
1F58- 21 25 29 2D 31 35 39 3D
1F60- 22 26 2A 2E 32 36 3A 3E
1F68- 22 26 2A 2E 32 36 3A 3E
1F70- 23 27 2B 2F 33 37 3B 3F
1F78- 23 27 2B 2F 33 37 3B 3F
*
```

Listing 3. YTABLE.

SCANA

In Listing 1 you'll find a program I call SCANA. Its purpose is to provide for saving anything that's on the hi-res screen into block shapes. Don't forget POKE103,1:POKE104,64:POKE16384,0 in a Hello program that runs before SCANA.

TESTTB (Listing 4) is necessary to have on disk when running SCANA. To enter TESTTB into your computer CALL-151 <RETURN> and type 800:, then enter the hex codes. BSAVE as TESTTB,A2048,L140.

SCANA is easy to operate, as long as game paddles are available. Before you run this program make sure you've put something on screen 1. Hit

the space bar upon entry, unless you'll be loading some other hi-res shape (a vector shape or a block shape you want to save only part of). Then use option 7 to paddle-define the shape block as directed. Use the paddles as directed, jot down height and width, and then answer Y for yes, the rectangle is okay. Answer N for no to the next question, unless there are more shapes on the screen requiring paddle-defining.

Now use option 9 to save your block shape table (which starts with shape 1 at $900 and increments one page, $100, per shape number). For the shape number question give the number for the current shape. You'll enter a shape name, and then enter option 1 or 2, depending upon whether you've saved any shapes more than one page long and whether your last shape defined had the highest shape number or not. You'll see the table length after BSAVEing is done. The table address is 2304 ($900). Anything that can be placed on hi-res screen 1 can be saved as a block shape with SCANA, as long as it doesn't use up over 5,888 of the 7,680 bytes on the hi-res screen.

Two-Page Flipping

Now let's look at a quick and easy example of what two-page flipping animation is all about. See Listing 2, called Two-Page Animation. You'll need to POKE103,1:POKE104,96: POKE24576,0 in Hello and RUN HELLO before you run this program. This procedure changes start-of-program POKEs so the Applesoft program is out of the way of both hi-res graphics page 1 ($2000–$3FF8) and page 2 ($4000–$5FF8).

Two things are necessary for two-page flipping animation: old lines must be erased once new lines have been drawn, and erasing and drawing must happen out of sight, so only completed lines are displayed, not presently-drawing lines. The latter is accomplished by displaying one page while drawing on or erasing from the other. POKE a 32 into 230 to draw on page 1; for page 2 POKE 230,64. POKE a 0 into −16300 to display page 1; for page 2 POKE −16299,0.

Here's the action sequence:
1) draw 0 on 2 but display 1 (preparation)
2) draw 1 on 1 but display 2 (preparation)
3) erase 0/draw 2 on 2 but display 1
4) erase 1/draw 3 on 1 but display 2
5) erase 2/draw 4 on 2 but display 1
6) erase 3/draw 5 on 1 but display 2
ad infinitum

The numbers immediately after DRAW or ERASE are shape numbers; all other numbers are hi-res screen page numbers.

Table Look-Up

Last, we'll investigate *table look-up*. Examine the hi-res page mapping on page 21 in your *Apple Reference Manual*. Note that \$2080 is the base address of the ninth line down at Y coordinate 8 (the first line was at Y = 0, remember). Now look at the ninth byte in the page 1 *high byte* section of YTABLE (Listing 3)—at addresses \$1EC0–\$1F7F. It is \$20, and the *low byte* section has a ninth byte of \$80—see addresses \$1E00–\$1EBF. Looking up these two numbers is all the base calculation it takes to figure screen address bases for any Y coordinate.

Confused? Okay, I'll explain it piece by piece. The numbers in the low byte table in \$1E00–\$1EBF are the low bytes of the screen addresses corresponding to the Y coordinates 0–191. The low bytes work for either page 1 or 2. The high bytes for page 1 are from \$1EC0 to \$1F7F and they start with \$20 (as in \$2000) and end with \$3F. Page 2 high bytes are from \$1D00 to \$1DBF and start with \$40 (as in \$4000) and end with \$5F.

Base address means the address of the specified Y coordinate line at the byte #0 column, which means that up to 39 may be added to that address to get the address of any of the 40 horizontal coordinates to be found along that Y coordinate line. Or in other words, to find the exact hi-res address of any byte, find the base address corresponding to its Y coordinate and then add the X coordinate as the displacement (to be added to this base address) to determine the final memory address. See Listing 5

```
0800- 00 00 00 07 CA E0 00 F0
0808- 04 C8 4C 04 08 98 85 FB
0810- A9 00 85 FA A5 FD 85 06
0818- A2 00 A0 00 20 11 F4 A4
0820- FE A2 00 B1 26 81 FA 88
0828- 18 E6 FA D0 02 E6 FB C0
0830- FF F0 04 C4 FF B0 EC C6
0838- 06 A5 06 C9 FF F0 04 C5
0840- FC B0 D5 60 A0 09 A6 07
0848- CA E0 00 F0 04 C8 4C 48
0850- 08 98 85 FB A9 00 85 FA
0858- A5 FD 85 06 A2 00 A0 00
0860- 20 11 F4 A4 FE A2 00 A1
0868- FA 51 26 91 26 88 18 E6
0870- FA D0 02 E6 FB C0 FF F0
0878- 04 C4 FF B0 EA C6 06 A5
0880- 06 C9 FF F0 04 C5 FC B0
0888- D3 60 00
```

Listing 4. TESTTB.

	Hex Bytes			Binary Bytes		
1)	7F	00	00	1111111	0000000	0000000
2)	7C	03	00	0011111	1100000	0000000
3)	70	0F	00	0000111	1111000	0000000
4)	40	3F	00	0000001	1111110	0000000
5)	00	7E	01	0000000	0111111	1000000
6)	00	78	07	0000000	0001111	1110000
7)	00	60	1F	0000000	0000011	1111100
1)	00	00	7F	0000000	0000000	1111111

Table 1. Shape sequences.

```
0320-  A4  06   LDY  $06
0322-  B1  CE   LDA  ($CE),Y
0324-  85  26   STA  $26
0326-  A5  E6   LDA  $E6
0328-  C9  40   CMP  #$40
032A-  D0  07   BNE  $0333
032C-  B1  DE   LDA  ($DE),Y
032E-  85  27   STA  $27
0330-  60       RTS
0331-  EA       NOP
0332-  EA       NOP
0333-  B1  1E   LDA  ($1E),Y
0335-  85  27   STA  $27
0337-  60       RTS
```

***Listing 5.** Machine language/assembly language YTABLE-using subroutine.*

for a machine language/assembly language YTABLE-using subroutine.

In this routine, $1E and $1F contain the low and high bytes of the page 1 high byte table *starting address*. $CE and $CF contain the low and high bytes of the low byte (either hi-res page) table *starting address*. Finally, $DE and $DF contain the low and high bytes of the page 2 high byte table *starting address*. With our present YTABLE address, $1E and $1F would contain $1EC0 ($C0 in $1E and $1E in $1F), $CE and $CF contain $1E00, and $DE and $DF contain $1D00.

In the above disassembled listing, if $40 (64) is found in $E6 (230), then page 2 is being drawn on, so we index into $1D00 (the address stored at $DE and $DF) the amount of the Y displacement. If $20 is found in $E6, then we end up at address 0333 in the listing indexing into $1EC0, the address stored in $1E and $1F.

As you can see, $6 gets loaded with the vertical (Y) coordinate in 0320 before anything else happens. We index into the low byte table to get the base address (low), and stick it into $26 (which is also where HPOSN puts it after base address calculation). The base address (high) gets stuck into $27 (where HPOSN would have put it) after indexing into either the page 1 (0333) table of high bytes or the page 2 (032C) table of high bytes.

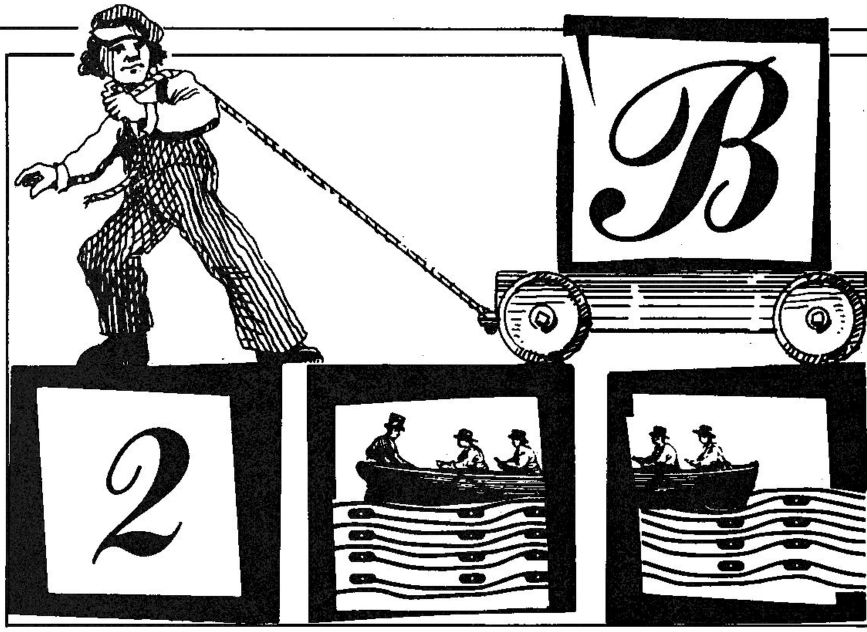
The reason for calculating a base address (and then adding X coordinate displacement) is to allow block shapes to be placed wherever we want on the screen. We know what shape table address to pull shape data from, but how will we know which hi-res page address to stick this shape-byte data into? All we have is X and Y coordinate information. The YTABLE method or the Applesoft HPOSN routine provide for hi-res address calculation using only X and Y coordinates to calculate from. We need either HPOSN or YTABLE routines every time we move up a line when we're loading shape table bytes (via TESTTB) onto the hi-res screen.

Consulting Figure 1 again, once we've loaded up the lower right, then lower left block shape byte (and, of course, one base calculation preceded the very first byte-load), we do a base calculation to get the hi-res base address of the bytes in the higher lines in the shape. The flying saucer would require eight base calculations.

The reason I put the short YTABLE-using algorithm from $320 to $337 is that it's a convenient out-of-the-way place for short routines or tables. The reason the YTABLE itself is at $1D00 is to provide room for 20 block shapes in memory ($900–$1CFF) and yet not interfere with

either the hi-res pages or high memory. YTABLE should be typed into the monitor at $1D00. BSAVE YTABLE at A7424,L640.

When using YTABLE from assembly, substitute JSR HPOSN with JSR $320 after LDA with the current Y coordinate and store (STA) that in $6. If you use YTABLE with TESTTB, notice that $6 *already* contains the current Y coordinate. To disassemble TESTTB for examination, use CALL-151 <RETURN> and 800L <RETURN>. Keep hitting L.



A Look Ahead

Next month we'll look at graphics and sounds together. The Apple has less sophisticated standard sound apparatus than the Atari or Commodore 64, but there *are* various peripheral boards with sound synthesizers available for the sound connoisseur. These include boards with arcade sound chips, music-oriented boards, and voice synthesizing boards like the Echo. ■

by Bill O'Brien

The Peripatetic Program

Not too long ago, a friend of mine, David Stone, started a book on telecommunications and microcomputers. Inevitably, we sat down together, he having bribed me with a sumptuous Petruccelli bologna on white with mayo, and we talked, from both a hardware and software point of view, about the different options available.

Most hardware is transportable from machine to machine. Let's face it, an RS-232 modem is an RS-232 modem, no matter what it's connected to. The only difference is a slight twist of the wrist to match up the output and input pins.

But software! Even among machines with the all-forgiving CP/M operating system, there are sometimes problems with what is essentially a good product. Having found something well done for one machine, it's in the Almighty's own hands whether or not it will work on an Apple (or a TRS-80 or a Commodore, if you'll forgive the references).

Once the ball started rolling, we went beyond just telecom programs. There have been quite a few times when, seeing a nice little ditty of a Basic program on some other machine, I've thought it would be nice to have a version for the Apple (or vice versa). Have you ever tried to do that?

Let me save you the trouble. In most cases it's a gigantic pain in the peripherals. Some must do it, though. After all, Remington Steele has both Apples and TRS-80s in her office, and they must have something in common, mustn't they?

Parsing for Programs

First you have to get rid of all the machine-specific statements. Line by agonizing line, out come the CLSs and the HOMEs to switch them around. Then you shuffle all the VPOs and HPOs and the PRINT @s. Let's not forget that all the INKEY$s and GETs and ON KBDs also have to be removed and reinserted. Tedious, to say the least.

After David had left to play with all the neat stuff he was going to write about (that's one of the plusses with being an author—you've got so much to fool around with), I sat down and thought about the problem.

Now, I am surely not the first one to contemplate the standards involved in writing good Basic code. In fact, Lou, another of my friends and a programmer par excellence, made that abundantly clear. It seems that these days the only way a programmer can make any money is to write a zillion or so programs a day. And, while, in the main, they're all "customized" to a particular application, on closer inspection quite a few lines look suspiciously identical, even down to the line numbers. There has been a movement in the last four or five years to concentrate programming routines into individual sections, or modules, that can be easily replaced without affecting the majority of the program.

Give Me Sections or Give Me a Break

Rather than going off into the world of programming theory, let's do something real, like an inventory. I would have said "integrated accounting system," but no one's really written a good one yet (oh boy, here come the letters!).

We'll keep it small and tight—just the basic information we need to know. That will include stock number, supplier, wholesale price, retail price, quantity on hand, minimum reorder quantity, and date of last order.

Refer to Listing 1. We'll start our modular coding with this convenient sequence. It's our data entry mask. By the way, when you see a line number and the letters REM after it, what follows is a REMark or comment from the person who wrote the program. It's ignored by the computer.

The mask routine assigns titles for the data we need to an array, LNE$, which has a total of seven elements. The length of each of these elements is 21 characters (the characters between the quotes, including the trailing space). We've done this, even though we didn't need so much space in all cases, so that all of the data starts at the same spot. The numbers leading each field will be used to reference the line, in case we've made a mistake somewhere and have to go back and make a correction.

The second array we're using is a numeric array, LNE. If you think back a few columns, you'll remember that the "$" makes a string array. LNE and LNE$ are separate and distinct from each other, somewhat like a quart of ice cream and a quart of pickles. In size they're the same, but in content and recognition they are totally different. LNE will hold the value for the maximum length we expect any of our data to be.

To increase portability, let's add a line at the beginning of the program:

```
10 DIM LNE$(7),LNE(7)
```

Some computer systems (ours included) allow the use of an array of under

---

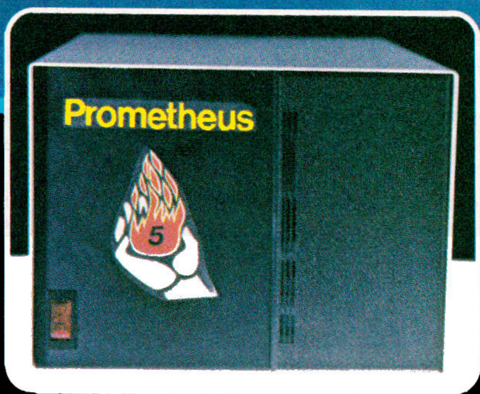
Address correspondence to Bill O'Brien at WABASA Consulting and Management, 111 Brook St., Scarsdale, NY 10583.

World Wide Data Systems Proudly Presents



Prometheus

External Winchester Hard Disk Drive Systems



The Prometheus family of external Winchester Systems were designed to put the power of hard disk data storage within the reach of all systems owners—without sacrificing performance, quality or reliability. All Prometheus Systems feature:

- Complete—Ready-to-run
- One year limited warranty (parts/labor)
- Error code correction / CRC generation / Verification
- Automatic micro-processor self test on power up. 12,000 hours mean time between failure
- Commercial power supply with automatic current limit and overload protection
- International AC input (All units easily accept 100, 120, 220 or 240 VAC, 47-440 HZ) Corcom RFI filter with transient suppression
- Error code—LED—alerts operator of system fault
- Expandable to 40 M Bytes

In addition each Prometheus System includes DOS+ 4.0a for the TRS80® Model III, and patches for Apple DOS, CP/M, Pascal, and MSDOS; and CP/M86 for the IBM Personal Computer.

PROMETHEUS - 5 $1,495.00

Also available 10, 15 & 20 megabytes hard disk systems for Apple, TRS80 & IBM-PC systems.

Circle 177 on Reader Service card.

Prometheus

Call or write:

Worldwide Data Systems, Inc.
17321 El Camino Real
Houston, Texas 77058
713/488-8022

```
20000   REM SCREEN MASK FOR DATA ENTRY
20010   LNE$(1)="1.      STOCK NUMBER: ":LNE(1)=15
20020   LNE$(2)="2.          SUPPLIER: ":LNE(2)=25
20030   LNE$(3)="3.         WHOLESALE: ":LNE(3)=8
20040   LNE$(4)="4.            RETAIL: ":LNE(4)=8
20050   LNE$(5)="5.           ON HAND: ":LNE(5)=5
20060   LNE$(6)="6.       MINIMUM QTY: ":LNE(6)=5
20070   LNE$(7)="7.   LAST ORDER DATE: ":LNE(7)=6
20080   REM DATA STARTS AT POSITION 22
20090   RETURN
```

Listing 1. Data entry mask sequence.

```
20100   REM DEFINITION SECTION
20105   BELL$=CHR$(7):REM RING THE BELL
20110   CLS$=CHR$(28):REM CLEAR THE SCREEN
20115   CLIN$=CHR$(30):REM ERASE A LINE
20120   CEOL$=CHR$(31):REM CLEAR TO END OF LINE
20125   CLEFT$=CHR$(8):REM MOVE CURSOR LEFT
20130   CRIGHT$=CHR$(9):REM MOVE CURSOR RIGHT
20135   CUP$=CHR$(11):REM MOVE CURSOR UP
20140   CDWN$=CHR$(10):REM MOVE CURSOR DOWN
20145   ULEFT$=CHR$(12):REM HOME CURSOR
20150   BS$=CLEFT$+" "+CLEFT$:REM BACKSPACE
20160   SLEN=24:REM SCREEN LENGTH
20165   SWID=80:REM SCREEN WIDTH
20170   ULIN$="_":REM UNDERLINE CHARACTER
20175   RETURN
```

Listing 2. Definitions sequence.

```
20200   PRINT CLS$;:RETURN:REM CLEAR THE SCREEN
20300   PRINT ULEFT$;:RETURN:REM HOME CURSOR
20400   REM DRAW DATA LINE
20410   FOR L=1 TO LNE(X)
20415   IF X=7 THEN PRINT"__/__/__";:L=LNE(X):
           GOTO 20430
20420   PRINT ULIN$;
20430   NEXT
20440   RETURN
20500   REM POSITIONING THE CURSOR
20510   PRINT CHR$(26);CHR$(COL);CHR$(ROW);
20520   RETURN
20600   PRINT CEOL$;:RETURN:REM CLEAR TO END OF LINE
20700   PRINT CLIN$;:RETURN:REM CLEAR TOTAL LINE
20800   REM RETURN TO BEGINNING OF DATA LINE
20810   FOR L=LEN(ENTRY$) TO 1 STEP -1
20820      PRINT CLEFT$;
20830   NEXT
20840   RETURN
20900   REM GET A CHARACTER
20910   PRINT"^";CLEFT$;:GET A$:PRINT " ";CLEFT$;:RETURN
21000   PRINT USING"#####.##";VAL(ENTRY$);:RETURN
```

Listing 3. Performance sequence.

Listing 4. Alphanumeric data entry sequence.

```
5000    REM MIXED NUMERIC AND STRING
5010    GOSUB 20900:REM GET A CHARACTER
5020    IF A$=CHR$(27) THEN EFLAG=1:RETURN
5030    IF A$<>CLEFT$ THEN 5070
```

Listing continued.

11 elements (0 to 10), without having it predefined in a DIMension statement, like line 10. Some of them won't. We add line 10 to be sure.

Doing It by the Book

Take a look at the screen control codes in Table 1, just in case you misplaced your standard device drivers manual thinking, "Oh, I'll never need *that*!" Then let's set up a definition section (Listing 2).

Not all computers have a Bell capability. Otherwise, we'd set the BELL$ = " ". Also, CHR$(9) on some systems is the TAB character. We've created a backspace (erase string) in line 20150. On some systems this is unnecessary and on the III it is nondestructive.

Some definitions we'll use directly, others we'll refer to through GOSUB statements. The REM statements we're including here may or may not be included in a real program, depending on the memory constraints. If you do program for anyone but yourself, it might be wiser to leave them out (to show how efficiently you use memory?) and record their values on paper with the rest of the documentation. It's a good idea to document for yourself, too. You never know what a few months will make you forget.

Now we need some performance lines, to do the task that is required. See Listing 3.

Since some machines require multiple steps to achieve what we can do on our III with but a single character, we've set up lines 20200 to 20840. The For...Next loop in lines 20410–20430 is used in printing the data mask. After each of the data labels is printed, the head drops down, and, using the value found in the element of LNE that corresponds to the same mask element, draws a line.

Line 20510 is important for many of the applications that require cursor positioning on the III (even CP/M applications). The III needs a lead-in character, CHR$(26). The next two characters are the horizontal position and the vertical position. Some machines want the vertical first, but *not* the Apple. Also, there are no inter-

mediary characters and no trailing characters. The column position is carried down to line 20510 in the variable COL, and the row position by ROW.

Line 20800 is a little trick that may or may not be of use. If, in the middle of entering some data, we decide to type in something else, we can flag the machine to come down and erase the line. The string ENTRY$ temporarily holds the value we're entering.

But, before we deal with that, lines 20900–21000 handle character entry, since there are *so* many ways to take a keyboard character. In some cases, depending on the type of data, it's easier to enter it character by character for a little more control. Since we also need a cursor character, I've elected to use the "Λ". No special reason—it just looks cute.

Since not all machines have a formatted output function (the PRINT USING command we Apple III people take for granted), that is relegated to a separate section. (We do have to go out of our way to make allowances for inferior machines!) Then, if needed, a routine can be inserted to actually tear apart the variable and restructure it.

Getting It All

We have various kinds of information to take in, and probably no one way handles them all. Let's say we have four different types:

- *String* data, which may or may not include numbers.
- *Numeric* data, only numbers.
- *Dollar* amounts, which have to be formatted for five spaces for dollars, a decimal point and two spaces for cents.
- *Date*, which we're going to accept in the form mm/dd/yy and check for validity.

It may seem silly to worry about the data entry on this level, but we must assume that whoever is using the program is going to do his or her utmost to blow it up. And you'd be surprised—sometimes the person who wrote the program can do it also.

Our data entry section starts at line 5000. Find it in Listing 4. If the key pressed in line 5020 is the escape key, CHR$(27), then we set a marker (EFLAG) and return. We'll use the escape key as a bail-out key, when we want to stop. Should we try to erase a character that isn't there—that is, if our temporary string, ENTRY$, is empty—the character is ignored and we go back to look for another. Some machines have problems dealing with strings whose lengths are 0, so we handle it (line 5040) without looking at the length as a comparison to anything.

What Goes In May Come Out

Erasing a character is more complicated than simply walking back over it. If we were using an Input statement, the machine would accomplish the task for us, but since we're not we have to do it. First we trim ENTRY$ back a little by making it equal to itself, but one character shorter (line 5060). I'm presuming you're familiar with the LEFT$ function in Basic. In short, it marks off the length of a string variable such as ENTRY$, starting at the left and counting up to the number of characters we've selected. In line 5060 we specify the new variable to be one character less than the current length of ENTRY$.

Line 5070 says that, if we hit the return key, CHR$(13), the program should return to where it came from. Lines 5080 and 5090 make sure of that tricky section we set up. If we press both the control and C keys, CHR$(3), we erase the line we're typing, position the cursor at the line's starting point, repaint the line, set ENTRY$ equal to nothing, and start all over again.

Lines 5100 and 5110 are limiters of a sort we'll see in some of the other data entry sections. The first one checks to make sure we haven't entered any control characters. Notice we've done this *after* we've checked for the backspace character,

CHR$(8); the escape key, CHR$(27); and the return key, CHR$(13). The second is something we may or may not want to use, depending on the application. It converts lowercase letters to uppercase, effectively removing the need for the alpha lock key. Again, in this type of program it really doesn't matter, but under some conditions we might want to include lowercase.

Finally, we print the character we've carefully accepted, add it to the current value of ENTRY$ and check to see if ENTRY$ is now as long as the maximum value we chose in LNE(X). If it is, we return to where we came from; if not we go back for another character.

Numeric entry follows the same pattern. See Listing 5. It looks the same, except for line 6100, which restricts the characters we can enter to the numbers between 0 and 9. Now we can jump right into the next routine in Listing 6.

Dollars and Sense

Well, Listing 6 looks pretty regular. Wait a minute! How about line 7120—and line 7130? It's pretty confusing right about there, isn't it?

What we've done here is assure ourselves that the dollar amount entered doesn't go above or below our expected figure. Line 7120 checks the figure in ENTRY$. If it finds that the string representation of the number we're organizing contains six characters, the Instr function makes sure that at least one of the characters is a decimal point. After all, if we're formatting for five dollar places, we really have no need for six.

Now, just in case there is a fractionalist in our midst, line 7140 keeps tabs on the numbers after the decimal. If it finds more than two—that is, if the decimal point happens to be the first character in the group composed of the last four characters, when it should actually be the third—we send it back for a retake.

Finally, we position the cursor at the beginning of the field, jump down to 21000 to print the formatted value, and check if we've reached the maximum length allowable. If so, we return to whence we came; if not, we

Listing continued.

```
5040    IF LEN(ENTRY$)=0 THEN 5010
5050    IF LEN(ENTRY$)=1 THEN ENTRY$="":PRINT BS$;:
          GOTO 5010
5060    ENTRY$=LEFT$(ENTRY$,LEN(ENTRY$)-1):
          PRINT BS$;:GOTO 5010
5070    IF A$=CHR$(13) THEN RETURN
5080    IF A$=CHR$(3) AND LEN(ENTRY$)=0 THEN 5010
5090    IF A$=CHR$(3) THEN GOSUB 20800:PRINT CEOL$;:
          GOSUB 20500:GOSUB 20400:GOSUB 20500:
          ENTRY$="":GOTO 5010
5100    IF ASC(A$)<32 OR ASC(A$)>126 THEN 5010
5110    IF ASC(A$)>96 AND ASC(A$)<123 THEN
          A$=CHR$(ASC(A$)-32)
5120    PRINT A$;:ENTRY$=ENTRY$+A$:IF LEN(ENTRY$)=
          LNE(X) THEN RETURN:ELSE GOTO 5010
```

```
6000    REM NUMERIC ONLY
6010    GOSUB 20900
6020    IF A$=CHR$(27) THEN EFLAG=1:RETURN
6030    IF A$<>CLEFT$ THEN 6070
6040    IF LEN(ENTRY$)=0 THEN 6010
6050    IF LEN(ENTRY$)=1 THEN ENTRY$="":PRINT BS$;:
          GOTO 6010
6060    ENTRY$=LEFT$(ENTRY$,(LEN(ENTRY$)-1):
          PRINT BS$;:GOTO 6010
6070    IF ENTRY$=CHR$(13) THEN RETURN
6080    IF A$=CHR$(3) AND LEN(ENTRY$)=0 THEN 6010
6090    IF A$=CHR$(3) THEN GOSUB 20800:PRINT CEOL$;:
          GOSUB 20500:GOSUB 20400:GOSUB 20500:
          ENTRY$="":GOTO 6010
6100    IF ASC(A$)<48 OR ASC(A$)>57 THEN 6010
6120    PRINT A$;:ENTRY$=ENTRY$+A$:IF LEN(ENTRY$)=
          LNE(X) THEN RETURN:ELSE GOTO 6010
```

Listing 5. Numeric data entry sequence.

```
7000    REM DOLLAR ENTRY
7010    GOSUB 20900
7020    IF A$=CHR$(27) THEN EFLAG=1:RETURN
7030    IF A$<>CLEFT$ THEN 7070
7040    IF LEN(ENTRY$)=0 THEN 7010
7050    IF LEN(ENTRY$)=1 THEN ENTRY$="":PRINT BS$;:
          GOTO 7010
7060    ENTRY$=LEFT$(ENTRY$,LEN(ENTRY$)-1):
          PRINT BS$;:GOTO 7010
7070    IF A$=CHR$(13) THEN RETURN
7080    IF A$=CHR$(3) AND LEN(ENTRY$)=0 THEN 7010
7090    IF A$=CHR$(3) THEN GOSUB 20800:PRINT CEOL$;:
          GOSUB 20500:GOSUB 20400:GOSUB 20500:
          ENTRY$="":GOTO 7010
7100    IF ASC(A$)<48 AND A$<>"." OR ASC(A$)>57 THEN 7010
7110    ENTRY$=ENTYR$+A$
7120    IF LEN(ENTRY$)=6 AND INSTR(ENTRY$,".")=0 THEN
          ENTRY$=LEFT$(ENTRY$,LEN(ENTRY$)-1):
          GOTO 7010
7130    IF LEFT$(RIGHT$(ENTRY$,4),1)="." THEN
          ENTRY$=LEFT$(ENTRY$,LEN(ENTRY$)-1):
          GOTO 7010
7140    GOSUB 20500:GOSUB 21000:
          IF LEN(ENTRY$)=LN(X) THEN RETURN:
          ELSE GOTO 7010
```

Listing 6. Dollar entry sequence.

get another character.

Friday Nights

Dates are very important. So important, in fact, that even though we're only taking in numeric data, we're still going to create a separate section to accept the date. This appears in Listing 7.

Yes, it is somewhat different. Everything's as you might expect up to line 8040, but from there on things get a little wild and crazy. Line 8050 relates to an action we take in line 8140. Let's deal with the last of the pair first.

In order to preserve the format we wish to collect the date in (mm/dd/yy), line 8140 forces the placement of a "/" after the second or the fourth character entered. That's great for holding the format in the forward direction, but what happens if we've made a mistake and want to erase some characters?

In some places there would be no problem at all, but when we back up enough to reach either of the "/"s, well, why get rid of a good format when we don't have to? So, when we reach the "/" and type a backspace, the commands in line 8040 make the cursor jump back one character before we use the destructive backspace (BS$). Our marker is preserved and the undesirable characters are deleted.

Lines 8080 to 8125 are the date validation section. The first thing checked for is the month. Since we want the month expressed as two digits, the first nine months of the year must start with a 0, and no month will ever begin with a number greater than 1. Hence, we have the program check for the first digit entered, and then, conditionally, the second.

We then make sure that if the month entered is February (02), the next digit is not larger than 2. Why? Well, I've never seen a February with more than 29 days.

Finally, just to assure ourselves that all the jokers in the crowd are taken care of, we check the full format of the date. If the date is the 31st of a month, we check to see if the month is valid. We also restrict the first digit of the date to numbers below 4, and, when a 3 is entered as the leading digit,the second digit is confined to numbers below 2.

We could go a step further and check for a leap year, but there are some things I'm willing to take on faith. If you care to add it, you have enough examples by now to do it yourself.

In fact, if you want to see a finished program, you're going to have to do it yourself. As I told you, we're not going to write a program here, just see how it can be done. (But, if you run into a problem, be sure to write. Between me and everyone else out there, somebody will know how to help.)

The Results

For a long time, a very long time, people have been mourning the intractability of Basic. It's a dumb language, complicated, hard to follow... ad infinitum. But it doesn't have to be. And that's what I've hopefully just shown. By breaking down the entry routines into modules that can be easily reached, not hidden (or should I say scattered) in the body of a program, we can increase the readability of the language.

Great, right? But, it's a bore to type the lines in all the time. Well, there's even relief for that. (Am I a great guy, or am I a great guy?)

If we were to type in OUTREC = 255 at the keyboard, we'd set the maximum size of a printed or displayed line to 255 characters (the III defaults to 80 characters). If we then typed in the module (just one at a time) wanted—the Date routine perhaps—and finished by typing

```
OPEN#1,"DATE.ASC"
OUTPUT#1
LIST
```

we'd have an ASCII, or text, file saved on disk.

```
8000    REM DATE ENTRY
8010    GOSUB 20900
8020    IF A$<>CLEFT$ THEN 8070
8030    IF LEN(ENTRY$)=0 THEN 8010
8040    IF LEN(ENTRY$)=1 THEN ENTRY$="":PRINT BS$;:
            GOTO 8010
8050    IF LEN(ENTRY$)=2 OR LEN(ENTRY$)=4 THEN
            PRINT CLEFT$;BS$;: ENTRY$=LEFT$(ENTRY$,
            LEN(ENTRY$)-1):GOTO 8010
8060    PRINT BS$;:ENTRY$=LEFT$(ENTRY$,LEN(ENTRY$)-1):
            GOTO 8010
8070    IF VAL(A$)=0 AND A$<>"0" THEN 8010
8080    IF LEN(ENTRY$)=0 AND VAL(A$)>1 THEN 8010
8090    IF ENTRY$="1" AND VAL(A$)>2 THEN 8010
8100    IF ENTRY$="02" AND VAL(A$)>2 THEN 8010
8110    IF MID$(ENTRY$,3,1)="3" AND VAL(A$)>1 THEN 8010
8115    IF LEN(ENTRY$)=2 AND VAL(A$)>3 THEN 8010
8120    IF LEN(ENTRY$)=3 AND RIGHT$(ENTRY$,1)="3" AND
            A$="1" THEN GOSUB 8500:IF DFLAG=1 THEN
            DFLAG=0:GOTO 8010
8125    IF LEN(ENTRY$)=3 AND RIGHT$(ENTRY$,1)="3" AND
            VAL(A$)>1 THEN 8010
8130    PRINT A$;:ENTRY$=ENTRY$+A$
8140    IF LEN(ENTRY$)=2 OR LEN(ENTRY$)=4 THEN
            PRINT "/";
8150    IF LEN(ENTRY$)=LNE(X) THEN RETURN:ELSE GOTO 8010
8500    REM CHECK THE DATE
8510    IF VAL(LEFT$(ENTRY$,2))=9 OR VAL(LEFT$(ENTRY$,2))
            =4 OR VAL(LEFT$(ENTRY$,2))=6 OR
            VAL(LEFT$(ENTRY$,2))=11 THEN DFLAG=1
8520    RETURN
```

Listing 7. Date entry sequence.

CODE	NAME	ARGS	FUNCTION
00	NUL		NO OPERATION
01	SOH		SAVE AND RESET VIEWPORT
02	STX		SET UPPER-LEFT CORNER OF VIEWPORT
03	ETX		SET LOWER-RIGHT CORNER OF VIEWPORT
04	EOT		RESTORE VIEWPORT
05	ENQ		TURN CURSOR ON
06	ACK		TURN CURSOR OFF
07	BEL		SOUND THE BELL
08	BS		CURSOR LEFT
09	HT		CURSOR RIGHT
10	LF		CURSOR DOWN
11	VT		CURSOR UP
12	FF		HOME CURSOR
13	CR		CARRIAGE RETURN
14	SO		TURN SCREEN OFF
15	SI		TURN SCREEN ON
16	DLE	MODE	SET TEXT MODE
17	DC1		SET NORMAL TEXT
18	DC2		SET INVERSE TEXT
19	DC3	COLOR	SET FORGROUND COLOR
20	DC4	COLOR	SET BACKGROUND COLOR
21	NAK	MOVE	CURSOR MOVEMENT CONTROLS
22	SYN		SYNCHRONIZE SCREEN
23	ETB	SHIFT	HORIZONTAL SHIFT
24	CAN	X	HORIZONTAL POSITION
25	EM	Y	VERTICAL POSITION
26	SUB	X Y	ABSOLUTE POSITION
27	ESC		NO OPERATION
28	FS		CLEAR VIEWPORT
29	GS		CLEAR TO END OF VIEWPORT
30	RS		CLEAR LINE
31	US		CLEAR TO END OF LINE

Table 1. Screen control codes.

(A brief aside here. We've opened a file on the default drive named DATE .ASC. OUTPUT#1 instructs the III to take all displayed information (anything PRINTed or LISTed) and send it out through the file we've opened as #1. By LISTing the module we create a file, consisting of the program lines, that can be handled the same as any text or ASCII file can be.)

What does that do, you ask? Ever hear of the Exec command? The proper format is EXEC <filename>, where the file chosen is in ASCII or text format. The III, having received the command, begins to process the named ASCII file. Each line is treated as a command typed in from the keyboard. If a line number is found, it sends that line to memory, as if we'd just typed in a program line.

If our file consisted only of program lines, then they would be accepted into memory and added to anything already there. Start to make sense? The only trouble would be if we hadn't used the OUTREC command. The format the III normally lists things in—80 characters, an indent, and any additional characters before the carriage return—causes a syntax error. By OUTRECing to 255 characters, we eliminate that indent character and remove the error.

If we've saved all of our modules, we can EXEC in all the ones we need and just fill in the program that surrounds them. It's as simple as that. No fuss, no muss. We write them only the first time. Once written and saved, they become a library of procedures (I bet all you Pascal people out there were waiting for me to say that) ready to call upon.

Imagine a phone number. There are a zillion problems with achieving the (xxx) xxx-xxxx format. Try to write the procedure as an exercise. Think of it in terms of your investing some time now and saving a heck of a lot of it later.

Thus, we come to the close of another month. Seventy-six more and I'll be equal to a III (80 columns?). In the meantime, live long and progam.

Ciao bene, AppleAmerica! ■

RH ELECTRONICS, INC.

COPYRIGHT © 1981 — PATENTS PENDING

566 Irelan, Buellton, CA 93427

(805) 688-2047

8:00 TO 5:00 CALIFORNIA TIME

COMPUTER PRODUCTS

DESIGNING • MANUFACTURING
ELECTRONIC ENGINEERING

SUPER FAN II™

FOR YOUR APPLE II* COMPUTER

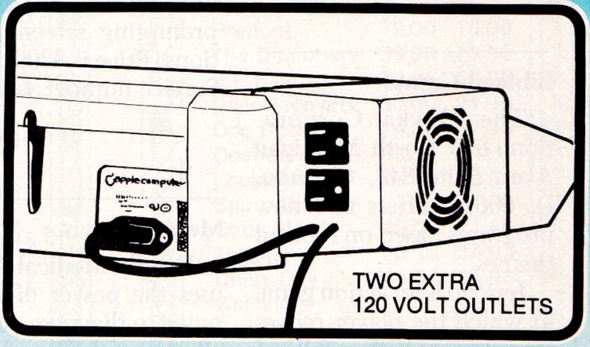


One Year Warranty

$74.95

With Zener Ray™ Protection $109.00

MASTERCARD — VISA



"COOL IT"

- ALSO FITS ON APPLE'S* NEW MONITOR STAND
- RED PILOT LIGHT ON/OFF SYSTEM SWITCH
- CLIPS ON — NO HOLES OR SCREWS • REPLACEABLE SWITCH
- AVAILABLE IN 120V or 240V AND 50/60 HZ • DURABLE MOTOR
- REDUCES HEAT CAUSED BY EXTRA PLUG-IN CARDS
- SOLD WORLD WIDE • UNIQUE 1 YEAR WARRANTY
- TAN OR BLACK COLOR • QUIETEST FAN ON THE MARKET
- INCREASED RELIABILITY — SAVES DOWN TIME AND REPAIR CHARGES
- LOW NOISE DUE TO DRAWING EFFECT OF AIR THROUGH YOUR COMPUTER AND SPECIAL FAN AND MOTOR DESIGN
- TWO EXTRA 120V OUTLETS FOR MONITOR AND ACCESSORIES TURN ON WHEN YOU TURN ON YOUR FAN (NOT AVAILABLE ON 240V MODEL)

SUPER FAN II™ WITH ZENER RAY OPTION $109.00

ZENER RAY™ TRANSIENT VOLTAGE SUPPRESSOR

OUR BUILT IN ADVANCED DESIGN UNIT GIVES
DRAMATIC COST SAVINGS — STOPS ANNOYING DOWN TIME
INSURANCE FROM VOLTAGE SPIKES - GLITCHES
DANGEROUS VOLTAGE SPIKES CAN JEOPARDIZE YOUR COMPUTER SYSTEMS
PROTECT COMPUTER - DISK DRIVE - PRINTER AND MONITOR

NO CUTTING WIRES • WON'T VOID WARRANTY, JUST PLUG IN SUPERFAN II WITH ZENER RAY

OTHER PRODUCTS BY RH ELECTRONICS, INC.

SUPER RAM II™ 16K RAM CARD FOR YOUR APPLE II. 2 YEAR WARRANTY $125

GUARDIAN ANGEL™ AN UNINTERRUPTABLE POWER SOURCE $595

12 VOLT TRANSVERTER 12 VOLT — RUNS YOUR APPLE II COMPUTER AND AND 5¼″ DRIVE FROM YOUR CIGARETTE LIGHTER $149

*Registered trademarks of Apple Computer Inc.

DEALER INQUIRIES INVITED

Circle 87 on Reader Service card.

New Software

edited by Tom Woods

Golf Game

An 18-hole computer simulation golf game for the Apple II is offered by the Hayden Software Company, 600 Suffolk St., Lowell, MA 01853.

Each hole is displayed on the screen from two angles. A "status line" shows the hole number, hole length, par number, the ball's distance from the hole and the number of strokes taken. Each player has 20 clubs from which to choose, including all popular irons, sand and pitching wedges, and woods. A player can compete against others or against par. Price is $24.95. Reader Service number 440.

Biblical Games

The Davka Corporation, 845 North Michigan Ave., Suite 843, Chicago, IL 60611, offers two new programs based on biblical themes.

Jericho is an action game in which the player relives the story of the fall of Jericho described in the Book of Joshua. A player who skillfully directs the blasts of his "ram's horn" can crumble the walls of Jericho. Bible Baseball is an educational game that tests one's knowledge of the history of the Old Testament. A keen knowledge of the Old Testament enables a player to triumph in a full-scale baseball game played against an opponent or against the computer. The Davka biblical series is designed for the Apple II. Jericho is priced at $29.95. Bible Baseball is $25.00. Reader Service number 442.

Stats Plus

Stats Plus, a new general statistics package with a powerful database management system, is available from Human Systems Dynamics, 9249 Reseda Blvd., Suite 107, Northridge, CA 91324.

Stats Plus is VisiCalc compatible. Electronic worksheet files can be used in preparing data files or in producing high-resolution graphics such as scatterplots, bargraphs, or polygon charts. A special set of database programs, Data Prep, is included in the package. The programs handle both random access and sequential files and produce data that can be instantly accessed using self prompting screen instructions. Price is $200. Reader Service number 441.

Medical Advisor

Family Medical Advisor uses the power of a computer to diagnose common ailments, obscure diseases, or simple childhood illnesses, using nearly 10,000 possible combinations of symptoms. Simple answers to a series of questions prompted by the Apple II Plus establish a unique data pattern. A special algorithm then enables the computer to analyze the accumulated data and to diagnose the most probable medical condition from a database of nearly 200 illnesses.

No medical or computer expertise is required, and medical terminology is defined on the monitor screen. Contact Navic Corporation, Box 14727, North Palm Beach, FL 33408. Price is $37.50. Reader Service number 444.

Basic Learning

Apex Software Company, 8781 Troy St., Spring Valley, CA 92077, offers an introductory program, Basic Learning Package, that teaches the fundamentals of programming in Apple's native programming language, Applesoft Basic. The program also shows the commands of the Disk Operating System version 3.3. Hardware required to run the program is an Apple computer with 48K of memory, Applesoft Basic in ROM, one disk drive with DOS 3.3, and a television or video monitor. Price is $49.95. Reader Service number 443.

Fortran 77

Softronics Inc., 6626 Prince Edward Place, Memphis, TN 38139, has released a Fortran 77 source program which provides Softrans protocol compatibility for file transfers between Apple computers and any host computer.

Softrans is used in conjunction with Softerm, a high-speed CRT terminal emulation program for the Apple. The Softrans protocol, which operates in a block mode using asynchronous communications, provides error detection with automatic retransmission, automatic binary encoding and decoding, CRC-16 checksum, and data compression to enhance line utilization. The Fortran 77 program is executed on the host system under the control of the Apple computer, which, when running the Softerm program, appears to the host computer as a standard CRT terminal. Price is $150. Reader Service number 446.

Wavy Navy

Sirius Software Inc., 10364 Rockingham Drive, Sacramento, CA 95827, offers Wavy Navy, a high seas battle game that features amusing graphics, sound effects, maritime music, and a variety of play options and levels. Price is $34.95. Reader Service number 447.

Prime Plotter

Primesoft Corporation, Box 40, Cabin John, MD, 20818, offers an integrated statistics package, Prime Plotter, designed for the Apple II Plus. The program has been designed as a modular system with built-in slots for plotting, statistics and plotter interfacing modules, as well as replaceable character sets, typeset designs and figure files. It also provides a replay feature, which makes the program suitable for preparing slide shows and presentations in which the graphics displays created by the user are reproduced. Price is $239.95. Reader Service number 448.

Shapes in Color

Shapes in Color is a Basic precision shape drawing

ARK COMPUTING INC.

Your Salvation In The Sea Of Inflation.

P.O. Box 2025
Corona, CA 91720

SPECIALS

CP/M

AMDEK 12" Amber Monitor	Microsoft Premium System Includes: softcard, ramcard, videx Videoterm, softswitch, user guide.	Novation Smart Cat 1200 Baud full duplex.
List: 210.00	List: 695.00	List: 595.00
ARK: 164.95	ARK: 525.00	ARK: 495.00

ASSEMBLERS

	List	ARK
A.L.D.S.	125.00	89.95
LISA	79.95	59.95
LISA ED PAK	119.95	79.95
MERLIN	64.95	54.95

COMMUNICATIONS

	List	ARK
ASCII Expresso Pro	129.95	99.95
DATA Capture 4.0	64.95	49.95
VISITERM	100.00	79.95
Z TERM PRO	149.95	109.95

DATABASES

	List	ARK
dBASE II	695.00	499.95
DB MASTER	229.00	159.95
DB Utility #1	99.00	69.95
DB Utility #2	99.00	69.95
PFS	125.00	89.95
VISIFILE	250.00	184.95

EDUCATIONAL

	List	ARK
Know Your Apple	34.95	29.95
Mastertype	39.95	29.95
Speed Reading Courseware	99.95	79.95
Typing Tutor II	24.95	19.95

FINANCIAL MODELING

	List	ARK
Calestar	145.00	109.95
Desktop Plan II	250.00	184.95
Visicalc	250.00	184.95

GRAPHICS

	List	ARK
The Artist	79.95	59.95
EZ Draw	49.95	39.95
GAS	75.00	54.95
Graphics Magician	59.95	44.95
Hires Secrets	125.00	89.95
Visiplot	200.00	159.95

HARDWARE

	List	ARK
Applicard 6mhz	595.00	395.00
Graphics Plus	149.95	99.95
Keyboard Plus	99.95	69.95
Kraft Joystick	64.95	49.95
Lower Case Plus	64.95	49.95
Lower Case Plus II	24.95	14.95
Micromodem II w/term. pak	409.00	319.95
Microsoft 16K Card	99.95	79.95
Microtek Parallel Interface	139.50	64.95
Videx Combo	425.00	250.00

MAILING LISTS

	List	ARK
Address Book	49.95	39.95
1st Class Mail	74.95	54.95
Magic Mailer	69.95	
Magic Mailer	69.95	54.95
MailMerge	250.00	189.95

PRINTERS

	List	ARK
C. Itoh Prowriter 8510	595.00	475.00
C. Itoh Prowriter 1550	995.00	750.00
C. Itoh Starwriter F10	1395.	1500.
NEC NEC 8023	695.00	475.00

UTILITIES

	List	ARK
Apple Doc	39.95	29.95
Applewriter Preboot	19.00	15.00
The Dictionary	99.95	79.95
E P F IV	79.95	59.95
Disk Recovery	30.00	21.95
Dos Boss	24.00	17.95
Dosource	39.95	24.95
Locksmith	99.95	69.95
Super Disk Copy III	30.00	21.95
Tip Disk #1	20.00	14.95
Utility City	29.50	21.95
Visicalc Preboot	49.95	39.95
Watson	49.95	39.95

WORD PROCESSORS

	List	ARK
Exec. Sec.	250.00	189.95
Magic Window	99.95	79.95
Magic Window II	149.95	119.95
Pie Writer	149.95	99.95
Screenwriter II	129.95	79.95
Super Text 40/80	175.00	129.95
Word Handler	199.00	149.95
Wordstar	495.00	369.95

IF YOU DON'T SEE WHAT YOU NEED, GIVE US A CALL AND WE'LL TRY TO GET IT FOR YOU!

CALL FOR FREE CATALOG
(714) 735-2250

We accept VISA/MASTERCARD, Personal checks (allow 10 days to clear) or COD (2.00 charge). Please include 3% for shipping (2.00 min.) or 5% for blue label (3.00 min.). Calif. residents add 6% sales tax. All items are new and carry manufacturers warranty. Prices and availability are subject to change without notice.

Circle 29 on Reader Service card.

program for the Apple II that enables the user to create and compile shapes that can be drawn on a medium-res grid in various colors, sizes and angles. It can be used to design shapes ranging from typography to animation. Various shapes can be designed in the high-resolution colors of green, violet, white, orange, blue and black. Price is $49.95. Contact Hayden Software Company, 600 Suffolk St., Lowell, MA 01853. Reader Service number 449.

Dream Machine

Psychological Psoftware, 4757 Sun Valley Road, Del Mar, CA 92014, offers a new program, Dream Machine, that can be used as a tool for understanding and evaluating one's dreams. The program includes a dream dictionary with over 200 listings, a dream log that allows the user to file dreams, and a 123-page illustrated book. Price is $49.50. Reader Service number 451.

Rhymes and Riddles

Spinnaker Software, 215 First St., Cambridge, MA 02142, offers a new educational software game for children designed for the Apple.

Rhymes and Riddles is a letter guessing game presented in three formats to help children spell and to learn words to nursery rhymes and popular sayings. The first format, Jokes and Riddles, asks the child to complete a joke or riddle by guessing a punch line and filling in a space. In Nursery, the computer screen shows blank spaces for the first line of a well-known nursery rhyme. When the child completes the first line, the second line appears. After all lines are completed correctly, the rhyme appears on screen with a color illustration and music. The third format, Famous Sayings, presents a child with a famous saying for completion. Rhymes and Riddles is priced at $29.95. Reader Service number 450.

SwebNeTProCo!

Southwest by Northeast Technical Products Corp. (SwebNeTProCo) has given ADC a new meaning with its Super-ADC (anything-to-digital) kit. The converter will interface your Apple II with up to eight objects. You simply respond to the prompt "Animal, Plant or Mineral?" and the Super-ADC does the rest. SwebNeTProCo says that customers have successfully interfaced their micros with shag carpeting, a plumber's helper, a wheelbarrow, three of the leading underarm deodorants, Jack Nicholson's hairline, the Ohio Turnpike, alienation in America and Montana. The kit costs $99.95 in unmarked bills and can be ordered from SwebNeTProCo, Cherry Blossom Lane, Unit 17, El Rancho Trailer Park, Wall, SD 57790. Reader Service number ???

Roadsearch

Columbia Software, 5461

Marsh Hawk Way, Box 2235, Columbia, MD 21045, offers a computerized road atlas for the Apple II. Roadsearch simplifies the process of determining driving routes, mileages, travel times and fuel usage. The program has a database of 406 cities and road intersections located in the United States and Canada. Also included in the database are 69,000 miles of interstate and major through highways. Price is $34.95. Reader Service number 452.

Portfolio Analyzer

C. R. Hunter & Associates Inc., 1527 Northwood Drive, Cincinnati, OH 45237, offers the Permanent Portfolio Analyzer, a new investment software program that allows the user to enter and analyze a portfolio of investments for a period of ten years. The user may select any of six scenarios: level inflation, rising inflation, runaway inflation, softlanding, deflation and uncertain. The program will then print a ten year projection analysis, projecting the portfolio's future purchasing power. Permanent Portfolio Analyzer runs on the Apple II, II Plus, and III. Price is $295. Reader Service number 453.

Kensington FORMAT II

Format II can perform almost any editing and formatting functions, and display them on-screen, exactly as they will print out—including paragraphs, page reaks, underlining, justification, etc.

Editing is a simple matter: single key edit formats text without resorting to impossible multiple-key editing. Key "D" is delete, key "E" is edit, "J" justifies and so forth. The commands, and its menu-driven arrangement, make even complicated text formatting easy for both professionals and the novice.

Subscripts, superscripts, underlines, shadow printing, boldface and other features are supported by most printers compatible with the Apple II.

Format II..... **$195.88** UPS DELIVERED

Accent
Graphic Solutions $121.88

Applied Software Technology
Versa Form $298.88
Versa Form/HD $380.88

Artsci
Magic Window $84.88
Magic Mailer $62.88
Magic Words $62.88
Magic Pkg. Combo $177.88
Magic Window II $121.88

Broderbund
Payroll $303.88
Receivables $303.88

Comshare
Target Planner Calc $47.88
Master Planner Calc $251.88
Applications Package $47.88
Planner Calc Package $77.88

Continental Software
Home Accountant $66.88
1st Class Mail $66.88
CPA #1-GL $195.88
CPA #2-AR $195.88
CPA #3-AP $195.88
CPA #4-Payroll $195.88
CPA #5-Proprty Mngmnt $377.88

Dakin 5
Programming Aid $77.88
Depreciation Planner $303.88
Budget Planner $121.88
Business Bookkeeping $303.88

Datamost
Tax Beater $106.88
R.E.I.P $106.88
Write-On $106.88
Personal Ledger $106.88
Market Technician $106.88
Property Manager $229.88

Datasoft
Datasm 65 2.0 .... $77.88
LISP Interpreter $103.88

Denver Software
Pascal Tutor $106.88
Pascal Programmer $229.88
Financial Partner $195.88

Don't Ask
S.A.M. Mouth $121.88

Eagle
Money Decisions $158.88

Franklin
Ace Writer $90.88

Hayden Software
Pie Writer $121.88
Pie Writer/Double $121.88
Pie Writer/Multi-80 .... $121.88
Applesoft Compiler Plus $84.88

DC Hayes
Terminal Program $84.88

Howard Software
Creative Financing $155.88
Tax Preparer 82 .... $121.88
Real Estate Analysis $155.88

Insoft
TransForth II $103.88
MatheMagic $77.88
GraphMagic $77.88
Prism $32.88
Mathe/Graph Magic $121.88

IUS
Datadex $121.88
Pro EasyWriter $140.88
TellStar II $69.88
FORTH Development $114.88
Pro EasyMailer $121.88
EasyMover $84.88
Datadex/HD $232.88
Pro EasyWriter/ EasyMailer $232.88

Kensingtom
Format II $195.88

Link Systems
Dataflex $158.88
Datalink $84.88
Link Index $121.88

LJK
Letter Perfect $121.88
Data Perfect $84.88
Edit 6502 .... $84.88

Lotus
Executive Briefing $158.88

Micro Lab
Invoice Factory $158.88
Merger $47.88
Learning System $121.88
VisiBlend $47.88
Visifactory $66.88
Asset Manager $158.88
Tax Manager $121.88
Data Factory $232.88
Wall Streeter $232.88
Payroll Manager $232.88

MicroPro
TASC Compiler $140.88
muLisp/muStar-80 .... $158.88
muMath/muSimp-80 .... $195.88
ADIOS muMath muSimp-8 .... $195.88
Multi-Plan (Apple DOS) $214.88

Z80 Power for Apple/Franklin



Microsoft Premium Package
Includes Z80 Softcard, 16K RAMcard, Videx Videoterm (with soft video switch and reverse video) & a user's guide... **$484.88**
The above, less 16K RAMcard ... **$449.88**

Ashton-Tate
dBase II (Z80) $529.88

CommSoft
Roots/M (Z80) $154.88

DJR Associates
FMS-80 (Z80) $691.88
FMS-81 (Z80) $289.88
FMS-82 (Z80) $289.88

Fox & Geller
Quick Code (Z80) $229.88
Dutil (Z80) $84.88

MicroPro
WordStar (Z80) $377.88
DataStar (Z80) $168.88
MailMerge (Z80) $76.88
SuperSort (Z80) $115.88
SpellStar (Z80) $115.88
CalcStar (Z80) $115.88
WS Customization $273.88
Word/Mail/Spell $636.88
WordStar/MailMerge $488.88
Word/Spell Star $488.88

MicroSoft
FORTRAN-80 (Z80) $155.88
A.L.D.S (Z80) $103.88
BASIC-80 Compiler (Z80) $303.88
COBOL-80 (Z80) $565.88
M-Sort 80 (Z80) $155.88
Multi-Plan (Z80) $214.88

Peachtree
PeachPak 40 (Z80) $414.88
40-Inventory (Z80) $282.88
40-Payables (Z80) $282.88
Peachtext (Z80) $350.88
ProofReader (Z80) $214.88
PeachCalc (Z80) $265.88
Mailing List (Z80) $265.88
PeachPak 80 (Z80) $414.88
80-Inventory (Z80) $282.88
80-Payables (Z80) $282.88
Text/Spell/Mail (Z80) $414.88

Prompt Doc
Manual Maker (Z80) $492.88

Quality Software
GBS (Z80) $492.88

Sorcim
SuperCalc (Z80) $229.88
SpellGuard (Z80) $155.88

Southwestern
ASCII Express (Z80) $107.88
Z-Term (Z80) $84.88
Z-Term PRO (Z80) $121.88
P-Term PRO (Z80) $107.88

Standard Microsystems
Quic-N-Easi AG (Z80) $210.88
Quic-N-Easi PRO (Z80) $278.88

Taurus
CP+ (Z80) $121.88

Muse
Data Plot $54.88
Address Book $47.88
Form Letter $84.88
Super Text 40/80 .... $140.88
Super Text 40/56 .... $103.88

Sierra/On-Line
General Manager $181.88

Select
Select WP $377.88

Silicon Valley
Word Handler $179.88
List Handler $79.88

SOF/SYS
Exec. Secretary $195.88
Exec Speller $66.88

Softech
UCSD P-system $473.88
BASIC & RunTime $177.88
Xenofile $47.88
Softeach $103.88

Software Publishing
PFS: Report $81.88
PFS $103.88
PFS: Graph $103.88

STC
Pro Time & Billing $303.88
Apartment Manager $251.88
Inventory $214.88

Stoneware
D.B. Master $180.88
D.B. Master (HD) $380.88

Sublogic
Graphics Combo $99.88

Systems Plus
General Ledger $288.88
GL/AP/AR $627.88
GL/AP/AR/Invent $782.88

VisiCorp
Desktop Plan $211.88
VisiPlot $171.88
VisiTrend/Plot $251.88
VisiDex $211.88
VisiTerm $90.88
VisiCalc 3.3 .... $211.88
VisiFile $211.88
Business Forcasting Model $90.88

Information & Orders
(603) 881-9855

Orders Only: (800)-343-0726

No Hidden Charges

FREE UPS shipping on all orders—No extra charge to use credit cards—All software shipped factory fresh & sold as is—No software returns acceptedt—COD orders accepted (please add $10 fee) No purchase orders accepted—No foreign or APO orders accepted—Minimum $50 per order—This ad was prepared in January: prices subject to change without notice

Our Computer Showroom is now open in Amherst, NH

Qume SPRINT 11 PLUS



$1539.88 UPS DELIVERED

- 40 characters per second, bi-directional, logic-seeking printhead action
- Fully-formed, 96 characters, 10, 12, 15 & proportional spacing
- 1/48 inch line spacing and 1/120 inch character spacing (minimums)
- Parallel or RS-232C interfacing includes cable (please specify)

The Franklin ACE 1000

The Apple-compatible Personal Computer



The Franklin **ACE 1000** includes 64K RAM, upper/lower case character set (40 columns), hi-res graphics, numeric keypad, 8 peripheral slots, joystick/paddle connectors, and a built-in cooling fan. The Franklin **ACE 1000** has full hardware/software compatability with all existing Apple II® products.

Our special package includes everything you need for a complete Z80 system (except the software). It includes: the Franklin ACE 1000 (with its DOS), two **Rana Elite 1 disk drives**, a **Rana Controller Card**, a **Pi-4** 9" Amber Monitor, the **Microsoft Premium Package** (includes Z80 card and Videx 80-column card with soft video switch and reverse video), a **Colverter** color option kit, a **Gemini-10** printer, a **PKASO** graphics interface card with cable and a box of **Elephant Memory Systems** 5¼" single-sided, single density diskettes (10 to a box).

All for only **$2943.88**, UPS delivered.

PRINTERS

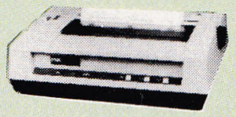
Anadex
Anadex DP-9501A **$1409.88**
Anadex DP-9620A **$1499.88**

C. Itoh
C. Itoh Prowriter **$499.88**
w/RS-232C **$609.88**
C. Itoh Prowriter 2 ...... **$734.88**
w/RS-232C **$789.88**
C.Itoh F-10 Starwriter, 40 cps
Parallel or RS-232C **$1499.88**
C.Itoh F-10 Printmaster, 55 cps
Parallel or RS-232C **$1799.88**
F-10 Series Tractor **$289.88**

Daisywriter
Daisywriter 2000 ...... **$1089.88**
Daisywriter Tractor **$149.88**
Daisywriter Cable **$49.88**

Diablo
Diablo 620 ...... **$1269.88**
Diablo 630 ...... **$1969.88**
Diablo 630 KSR **$2694.88**

TEC



DMP-85 Printer **$469.88**

Star Micronics



Gemini 10 ...... **$419.88**
Gemini 15 ...... **$539.88**

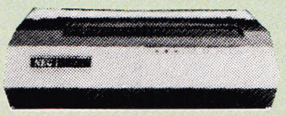
IDS
IDS Prism 80 ...... **$1104.88**
above w/graphics **$1189.88**
above w/sheetfeed **$1294.88**
above w/4-color **$1539.88**
IDS Prism 132 ...... **$1269.88**
above w/graphics **$1339.88**

PRINTERS

IDS Prism (cont)
132 w/sheetfeed **$1459.88**
above w/4-color **$1699.88**
IDS Microprism **$579.88**

Okidata
Microline 82A **$429.88**
82A Okigraph ROM **$44.88**
82A Tractor **$59.88**
82A Roll Paper Holder **$49.88**
Microline 83A **$689.88**
83A Okigraph ROM **$44.88**
Apple Okigraph II **$59.88**
Microline 84 w/graphics & tractor
Parallel, 200 cps **$1039.88**
Microline 92 ...... **$539.88**
Microline 93 ...... **$929.88**

NEC
NEC 3510 ...... **$1929.88**
NEC 3530 ...... **$1809.88**
NEC 3550 ...... **$2199.88**
3500 Tractor **$239.88**
NEC 7730 ...... **$2399.88**



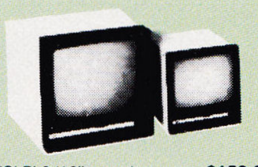
NEC PC-8023A **$509.88**

Qume
Qume Sprint 9/45 ...... **$2109.88**

Smith-Corona
Smith Corona TP-1 ...... **$579.88**
Specify either 10 or 12 cpi, & parallel or RS-232C interface

CALL FOR PRICES on Cannon, Centronics, Datasouth, Epson, Mannesmann Tally, Panasonic, Ricoh, Silver Reed, Transtar & others

MONITORS



USI Pi-2 (12" green) **$159.88**
USI Pi-3 (12" amber) **$189.88**
USI Pi-4 (9" amber) **$159.88**

Amdek
Amdek 300G (12" green) **$169.88**
Amdek 300A (12" amber) **$184.88**
Amdek Color I (Comp.) **$359.88**
Amdek Color III (RGB) **$479.88**
Apple RGB Interface **$159.88**

Electrohome
1302-1 (Medium RGB) **$329.88**
1302-2 (Hi-res RGB) **$579.88**
Super Color Option **$209.88**

NEC
JB 1210 (Comp.) **$364.88**

Zenith
ZVM 121 (12" green) **$139.88**

MODEMS

DC Hayes
Hayes 300 Baud **$234.88**
Hayes 1200 Baud **$569.88**
Hayes Micromodem II **$289.88**
above, w/software **$319.88**

Novation
103 SmartCat **$224.88**
103/212 SmartCat **$509.88**
Apple Cat 300 ...... **$334.88**
Apple Cat 300-1200 Kit **$324.88**
Apple Cat 1200 ...... **$579.88**
Cat **$159.88**
D-Cat **$159.88**
J-Cat **$139.88**

Multi-Line Junction Box **$39.88**

APPLE HARDWARE

PKASO Smart Interfaces
For Apple/Franklin centronics parallel output. Specify C.Itoh, Centronics, Epson, IDS*, NEC or Okidata printers.
PKASO card & cable **$139.88**
*IDS Color card & cable **$159.88**

Parallel Card w/cable **$89.88**

Corona Data Systems
5mB Hard Disk **$2049.88**
10mB Hard Disk **$2459.88**

MicroSoft
Premium Package—A 16K RAMCard, Z80 SoftCard, Videx VideoTerm w/soft video switch, & the
Users Guide **$484.88**
Z80 & Videx **$449.88**
Z80 SoftCard **$219.88**
16K RAMcard **$79.88**

Mountain Computer
CPS MultiFunction **$169.88**
RAMplus + 16K **$159.88**
RAMplus + 32K **$179.88**
ROMplus + ... **$139.88**
ROMwriter Card **$154.88**
5mB Hard Disk **$2364.88**
10mB Hard Disk **$2829.88**

Orange Micro
Grappler + ... **$149.88**
Bufferboard (16K) **$149.88**

Practical Peripherals
16K Microbuffer II **$219.88**
32K Microbuffer II **$249.88**

Rana Systems
Elite 1 Disk Drive **$299.88**
Elite 2 Disk Drive **$509.88**
Elite 3 Disk Drive **$659.88**
Disk Controller **$109.88**

Saturn Systems
32K RAM Board **$199.88**
64K RAM Board **$344.88**
128K RAM Board **$484.88**

SSM
AIO Multi-Function Card **$199.88**

Videx
VideoTerm **$239.88**
VideTerm Combo **$259.88**
Enhancer II **$114.88**

Vista
A-800 Disk Controller
for 8" disks **$399.88**
Vision 80 Card **$269.88**

Information & Orders
(603)-881-9855

Orders Only: (800)-343-0726

No Hidden Charges

FREE UPS shipping on all orders—No extra charge to use credit cards—All equipment shipped factory fresh with manufacturer's warranty—COD orders accepted ($10 fee added)—No purchase orders accepted—No foreign or APO orders accepted—Minimum $50 per order—This ad prepared in January: prices are subject to change.

HIGH TECHNOLOGY AT AFFORDABLE PRICES

THE BOTTOM LINE

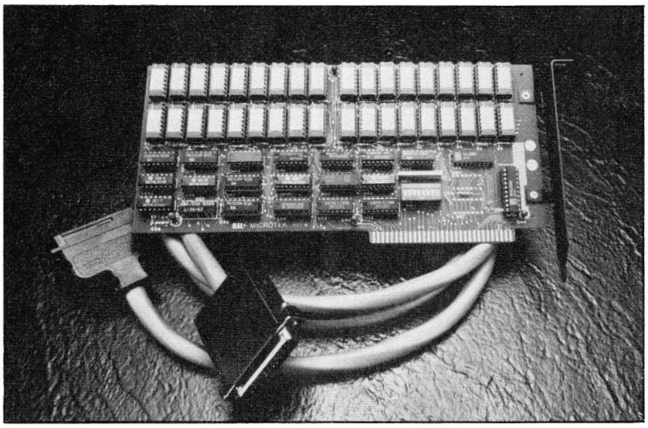
MILFORD, NH 03055-0423 □ TELEPHONE (603) 881-9855

New Products

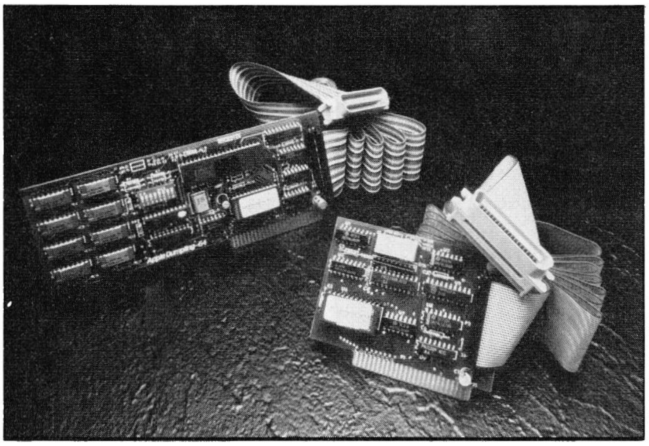
edited by Tom Woods

The Dumpling

Microtek Inc., 9514 Chesapeake Drive, San Diego, CA 92123, announces the Apple Dumpling series of parallel interface cards for graphics printers, presently available in two formats. The Dumpling GX is a non-buffered graphics interface. It interfaces with many graphics printers including the PMC, the Data South, the Mannesmann Tally and the new Apple printers. The Dumpling 64 is an expanded version of the Dumpling GX that permits user upgrade to 64K. The Apple Dumpling Series starts at $159. Reader Service number 455.



The Dumpling GX from Microtek Inc.



An expanded version of the GX, the Dumpling 64.

Multifunction Printers

Infoscribe Inc., 2720 South Croddy Way, Santa Ana, CA 92704, offers three new multifunction printers. The Infoscribe 1100 performs in a data/draft processing mode at 200 cps, in a correspondence mode at 100 cps, and in a business mode at 40 cps. It also produces dot-addressable graphics with a resolution of either 72 by 72 or 144 by 144 per inch.

The Infoscribe 1200 uses a four-color ribbon to produce up to eight colors for high text highlighting and graphics presentations. The unit prints at 200 cps in the draft/data processing mode, 100 cps in the correspondence mode and 40 cps in the business letter mode.

Infoscribe 1500 is a high speed model which prints at 400 cps in the draft/data processing mode and 200 cps in the correspondence mode. Dot-addressable graphics can also be produced in this unit.

Infoscribe 1100 is priced at $2295. The 1200 is $2495, and the 1500 is $2995. Reader Service number 451.

Kolor II

Synetix Micro Products, 15050 NE 95th St., Redmond, WA 98052, offers a single board product for the Apple II, the Kolor II, that provides an interface between the Apple II and video inputs such as recorders, cameras and live broadcasts. Utilizing the Texas Instruments TMS 9918, a video display processor, the Kolor II can display 16 colors with a resolution of 256 × 192 pixels that may be mixed with an external video input.

The Kolor II has four video display modes: Graphics I, Graphics II, Multicolor and text mode. Graphics information from the Apple computer is downloaded to the internal 16K or 32K memory of the Kolor II. All refresh and sync signals are produced by the Kolor II for composite video output for a video monitor, video tape recorder, or standard television set utilizing an rf modulator. The Kolor II plugs into any I/O slot of the Apple II or Apple II Plus and requires no external power. Price is $290. Reader Service number 457.

PipeLine

Interactive Structures Inc., 146 Montgomery Ave., Bala Cynwyd, PA 19004, offers an IS PipeLine printing buffer with random access printing. The IS PipeLine selects sentences, paragraphs, graphs or pictures from different programs to compose a finished document. Graphs can be inserted into reports and addresses put in form letters. Multiple copies are made automatically. The PipeLine also includes conventional FIFO operation; data is loaded into the buffer as fast as the computer can send it. PipeLine feeds the data out at the right speed for the printer.

Other features include: compression of data for efficient utilization of memory space, the ability to bypass buffer operations for straight-through printing, a simple erase feature to clear the buffer, an automatic duplication feature, and expandability from 8K bytes to 128K bytes.

The PipeLine is compatible with the PKASO Printer Interface for Apple computers. Price ranges from $195 to $495, according to memory capability. Reader Service number 454.

Microbuffer II

Microbuffer II is an interface card for the Apple II with up to 32K of on-board RAM for data buffering. It is slot-independent, fitting into any slot except #0 directly inside the Apple II. Microbuffer II features on-board firmware for text formatting and advanced graphics dump routines. Both serial and parallel versions have low power consumption.

ANOTHER DISCOVERY FOR orangeers

THE ORANGEPEEL™

POWERFUL. EXPANDABLE. AT $790, REMARKABLE.



Yesterday, buying a computer meant compromising. Powerful units carried powerful prices. Affordable units were little more than toys.

Today, there's the ORANGEPEEL. The powerful, expandable, highly versatile microcomputer priced at a remarkable $790.

- FULLY APPLE II® COMPATIBLE.
- LANGUAGE FLEXIBILITY — Modular design permits the use of all disc based Apple II software, CP/M,® UCSD-p.*
- MULTIPLE CPU's (as available).
- 76K RAM.
- DISC BASED — requiring standard Apple II drive.
- DETACHED KEYBOARD — Full ASCII upper and lower case.
- HIGH AND LOW RESOLUTION B/W GRAPHICS.
- 3 I/O SLOTS EXPANDABLE TO 7.
- ORANGE BUS FOR INTERNAL EXPANDABILITY.
- 6 MONTH REPLACEMENT WARRANTY — (Extended one year optional).

THE ORANGEPEEL COMPUTER. We anticipate the future.

See your local ORANGE dealer today.

orange computers inc

orange computers inc.
10-1495 bonhill road
mississauga, ontario l5t 1m2
(416) 677-7782

® Apple II is a registered trademark of Apple Computers Inc.
® CP/M is a trademark of Digital Research Inc.
*UCSD-p is a trademark of The Regents of The University of California.

Available in Canada, USA, Hong Kong, Singapore, Australia and Japan.

Circle 153 on Reader Service card.

Special functions include listing formatter, self-test, buffer zap, transparent and maintain modes. Price is $259. Contact Practical Peripherals Inc., 31245 La Baya Drive, Westlake Village, CA 91362. Reader Service number 458.

Video Cards

Microtek Inc., 9514 Chesapeake Drive, San Diego, CA 92123, offers two new video interface cards for the Apple and Franklin computers. The Magnum-80 is an 80-column video card which is completely Videx compatible and available with an alternate, built-in character set. The Rainbow-256 turns the Apple II/II Plus into a 256 color RGB driver for analog RGB monitors. Price is $269 for either card. Reader Service number 459.

Graphics Compatibility

A printer interface card for graphics compatibility between the Apple II and the Mannesmann MT 160 multi-function matrix printer is now available. With the interface card, priced at $149, and the NT 160L, priced at $990, the Apple user can reproduce his computer graphics on hard copy, print letter quality text for word processing duty and print reports at 160 cps. The interface consists of a single PC board and cable. The board inserts in the computer, the cable connects the computer and printer, and the user has a handshake for high resolution graphics and screen dumps. For further information contact Bill Clancy, Mannesmann Tally, 8301 South 180th, Kent, WA 98032. Reader Service number 460.

Speech Synthesizer

Multitech Electronics, 195 W. El Camino Real, Sunnyvale, CA 94086, offers a speech synthesizer board for the Apple II. The SSB-Apple can be used for language instruction, speech therapy, video games, and experiments in speech synthesis. The 2.75 × 6-inch board is based on Texas Instruments' TMS-5220 high-performance speech synthesis device. The board plugs directly into any space slot of the Apple II. Reader Service number 462.

Static RAM

Legend Industries Ltd., 2220 Scott Lake Road, Pontiac, MI 48054, offers an 18K Static RAM card for the Apple II, Apple II Plus, and Apple IIe. The card contains a battery backup system to allow memory retention after power-down, enabling the user to store programs and/or special monitor routines. The card draws very low power and can be used in any slot. A write-enable write-protect switch allows the user to protect the information in the card or alter the data at will. Price is $149.95. Reader Service number 461.

An April Special

When squads of aliens are diving or missiles are homing in, even the finest joystick or paddle may not respond quickly enough to save game players' bacon. Input Peripherals' new Joyspeak control device

works as a speech synthesizer in reverse, translating your spoken or shouted commands instantaneously into game action or response.

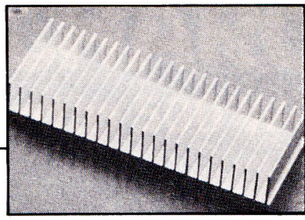
Holding the Joyspeak microphone (and leaving one hand free to flail the air), you simply give orders as would a captain on the bridge of a starship. Common command sequences such as "Left! Up! No! Fire! Over there! Back! Aaargh!" are immediately acted upon. A quick decision may save the day; a slip of the tongue can ruin a record score.

Price is $119.95 from Input Peripherals, 2230 Hoax Drive, Fictitious, NY 10022. Reader Service number 0000.

Investors Interface

Investors can collect stock data and base decisions on the analysis of their choice with Investors Interface from Marketware, Box 34647, Richmond, VA 23234. Investors Interface captures Dow Jones system stock data and converts it for use by the VisiCorp series, Appleplot and other software. Stock quotes are retrieved by specifying portfolios or stock symbols prior to logging. The Investors Interface requires an Apple II Plus with 48K, a Hayes Micromodem II and one or two disk drives. The package includes a tutorial manual with VisiCalc examples.

The key to the different interface possibilities offered by the Investors Interface is the Data Interchange Format (DIF). The package allows one to choose the exact stock data needed and converts it to DIF or Appleplot format. It is then ready to be used by other software. Daily Portfolio data can also be printed, displayed or converted to DIF format. Price is $125. Reader Service number 463.



The Heat Snatcher.

Heat Snatcher

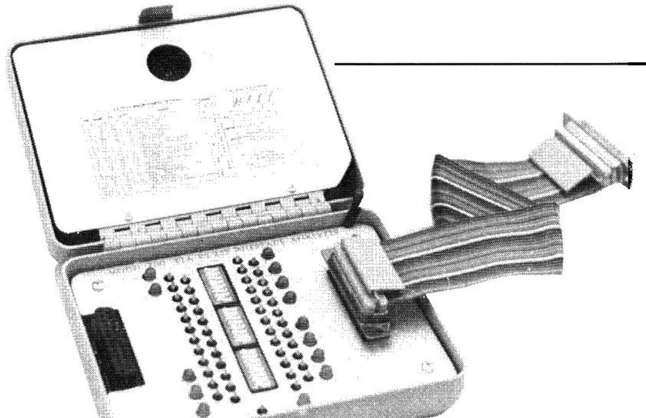
The Heat Snatcher is a passive cooling device that mounts on the power supply inside of an Apple or other type of microcomputer that uses an enclosed power supply. It operates without noise or power consumption and reduces the operating temperature of the power supply and within the computer cabinet. Heat Snatcher may be used alone or in conjunction with other devices. Price is $9.95. Contact Juli Company, 1415 S. Harlem Ave., Berwyn, IL 60402. Reader Service number 456.

A Basic Book

Basic Apple Basic is a guide to Applesoft Basic from beginning concepts to advanced topics. It provides alternative techniques for programming in Apple Integer Basic. More than 80 sample programs are included in the book, and many programs are divided into small segments that will fit on the Apple screen. Contact Hayden Book Company Inc., 50 Essex St., Rochelle Park, NJ 07662. Price is $12.95. Reader Service number 464.

Interface Analyzer

The Model 700 EIA RS-232 Interface Analyzer is a diagnostic tool designed for use at the standard EIA RS-232 or CCITT V.24 data interface of modems, multiplexers, terminals, and computers. It is inserted in series between the data terminal equipment and the data communications equipment to provide access to and monitoring of all data, timing and control signals.



The Model 700 EIA RS-232 Interface Analyzer.

The Model 700 uses tristate LEDs to display polarity, activity and validity of all key interface signals in red, green and red-green mixtures. It retails for $275 and is available from Electro Standards Laboratory, Box 9144, Providence, RI 02940. Reader Service number 453.

Cider Vinegar

There is said to be an ancient Chinese custom that the craftsman always weaves a couple of small errors into his work as an act of humility and to provide the viewer or user with the joy of discovery and the feeling of one-upmanship. (I'm not sure how one-upmanship is written in ideograph.)

inCider *has decided to embrace this venerable and honored custom in an ongoing way. We note that most of our authors are already cooperating, but to be sure, the editors will insert a couple of additional "bugs" in each article.*

The Cider Vinegar department will then chronicle our readers' successes at ferreting out inCider's *worms. The reader who can prove that he or she has found* all *the mistakes in a particular issue will be offered a computer-operated mousetrap as a modest prize. While, of course, the magazine has a complete master list of all the bugs in each issue, the editorial staff will divulge information from this list to no one.*

Thank-you to Lloyd Loring of South Bend, IN for suggesting "Vinegar" as the name of inCider's *errata section. Also to John Davidson of Marlow, NH (one of our authors) for pointing out* inCider's *roots in Chinese custom.*

Apple/80, January

In reading over my article "Apple/80" in the January, 1983 issue I noticed a few errors which crept in and need correcting. First, on page 102 Listing 2B is incomplete as printed. The following eight bytes should be added to the start of that code:

$11F8: A0 0A B1 69 0A 85 1E AA

The example cannot work without these eight bytes.

Second, on page 98 the first three bytes of the shape table are incorrectly listed. They should be:

$800: C0 00 80

They are, however, correctly printed in the body of the article.

Third, in Table 3 on page 107 the fourth example appears to have no translation. It is in fact the "GET K$" that mysteriously appears on the line above. Example three should end with "POKE −16368,0".

I also made one mistake. In Program Listing 4, the end of line 2 states "BLOAD APPLE-80 TABLE + PRINT". In the article I had suggested saving the binary file as "APPLE-80". Clearly these file names should match. Since "APPLE-80" is shorter, do as I say, not as I did. Shorten the end of line 2 to "BLOAD APPLE-80".

Hap Gaylord
PO Box 222
Bedford, NY 10506

New Products, February

In the February New Products column we incorrectly listed the price of the Z-80 Appli-Card, from Personal Computer Products Inc. The Appli-Card costs $295.

Screen Revelations, February

There is an error in my article "Screen Revelations" in the February issue of *inCider*. "255 numbers" in the tenth line of the third paragraph should read "256 numbers."

Winfield H. Edwards
300 Firwood Drive
Grants Pass, OR 97526

Fudge It!, February

We inadvertently omitted a piece of code from Listing 1 in February's "Fudge It!" column. At the bottom of page 152, line 1006 should read:

```
1006 HH = (BYTE + 7) - ( INT ( (BYTE + 7) / 256) * 256): POKE SOI + 4,HH: RETURN
```

The HH: RETURN at the end is the elusive part. Our thanks to Sidney Bowhill of the University of Illinois for alerting us to the omission.

Art Credit, February

The artist who created the border for Paul Payack's Mutually Unintelligible Response, on pages 82 and 83 of the February *inCider*, is Robert Dukette. Our apologies that his name was unreadable on the page.

TURBOCHARGE YOUR APPLE*

With The Quentin 500 Winchester Disk Drive



Introductory Prices

5 MB	$1995
10 MB	$2195
15 MB	$2395
20 MB	$2595

When you're ready to go the distance, the Quentin 500 is the high density, high speed Winchester subsystem that can fuel your Apple into a first-class powerhouse.

It's Deep. Store and retrieve 5, 10, 15 even 20 megabytes of formatted data on a single drive. With 20 megabytes of juice, you've got the equivalent of 140 conventional floppy disks, 4800 full pages of text, or all the financial files of a $50 million business.

It's Fast. The Quentin 500 accelerates from 0 to *Read* in 70 MS. And the DMA data transfer rate zips along at 5 MBits per second.

It's Accurate. The Quentin 500 moves out in front with the remarkable new Disk Drive Self Test Diagnostic System. Flick a switch and the 500 runs through its paces checking all Winchester operating functions, and verifying the integrity of the drive and the data. While the controller's full 32 bit Error Correction Code overrides all data field errors.

It's Reliable. Whether you're on-line all day, or driving in short, fast sprints, the Quentin 500 is strictly high performance. MTBF 10,000 power-on hours, with no preventive maintenance required.

Every Quentin 500 is delivered completely assembled, and is fully tested on Apple operating systems. And the Q-500 has a full one-year factory warranty.

It's Compatible and Apple-Beige. Plug it into your favorite Apple, and shift into full power. Software support packages provided to ensure complete compatibility with DOS, CP/M, Pascal and protected software.

And It's Very Affordable. The Quentin 500 is the first premium quality Winchester subsystem offered at a price all businessmen, professional corporations and serious computer users can afford.

Count On Quentin For Quality. Quentin Research has been designing and building mass storage subsystems for corporate users for five years. That's quite a few times around the track.

We've put those five years to good use by developing state-of-the-art disk technology for business and personal use, and by building a reputation for reliable products and stable management.

The Hotline to our Product Support Manager is open every week day for information on our full product line. When you're ready to expand your computer's capabilities, turbocharge with the Quentin 500. It's an information powerhouse.

QUENTIN RESEARCH, INC.
19355 Business Center Drive
Northridge, CA 91324
(213) 701-1006 • Telex 910-493-2174

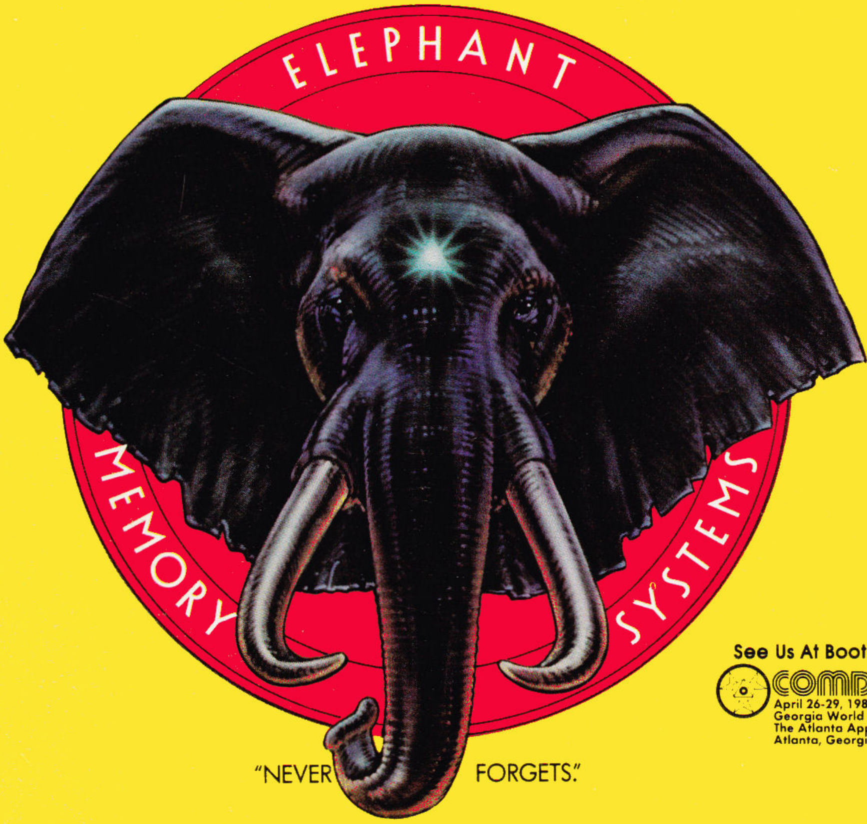
To Order Check, money order, Visa or Mastercard number. Calif. residents add 6% sales tax. Authorized club discounts available. Call or write for Quentin 500 Specifications and Features.

* Apple is a registered trademark of Apple Computer, Inc.

Circle 30 on Reader Service card.

Circle 30 on Reader Service card.

REMEMBER:



See Us At Booth #1146
COMDEX™/SPRING '83
April 26-29, 1983
Georgia World Congress Center and
The Atlanta Apparel Mart
Atlanta, Georgia

MORE THAN JUST ANOTHER PRETTY FACE.

Says who? Says ANSI.

Specifically, subcommittee X3B8 of the American National Standards Institute (ANSI) says so. The fact is all Elephant™ floppies meet or exceed the specs required to meet or exceed all their standards.

But just who is "subcommittee X3B8" to issue such pronouncements?

They're a group of people representing a large, well-balanced cross section of disciplines—from academia, government agencies, and the computer industry. People from places like IBM, Hewlett-Packard, 3M, Lawrence Livermore Labs, The U.S. Department of Defense, Honeywell and The Association of Computer Programmers and Analysts. In short, it's a bunch of high-caliber nitpickers whose mission, it seems, in order to make better disks for consumers, is also to make life miserable for everyone in the disk-making business.

How? By gathering together periodically (often, one suspects, under the full moon) to concoct more and more rules to increase the quality of flexible disks. Their most recent rule book runs over 20 single-spaced pages—listing, and insisting upon—hundreds upon hundreds of standards a disk must meet in order to be blessed by ANSI. (And thereby be taken seriously by people who take disks seriously.)

In fact, if you'd like a copy of this formidable document, for free, just let us know and we'll send you one. Because once you know what it takes to make an Elephant for ANSI . . .

We think you'll want us to make some Elephants for you.

ELEPHANT.™ HEAVY DUTY DISKS.

For a free poster-size portrait of our powerful pachyderm, please write us.
Distributed Exclusively by **Leading Edge Products, Inc.**, 225 Turnpike Street, Canton, Massachusetts 02021
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.